UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO CIÊNCIA DA COMPUTAÇÃO

AMIRTON BEZERRA CHAGAS

A Recommender System to Support the Development of Context-Aware Intelligent Transportation Systems

Recife

2022

AMIRTON BEZERRA CHAGAS

A Recommender System to Support the Development of Context-Aware Intelligent Transportation Systems

Trabalho apresentado ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, como requisito parcial para obtenção do grau de Doutor em Ciência da Computação.

Área de Concentração: Redes de Computadores e Sistemas Distribuídos

Orientador: Carlos André Guimarães Ferraz

Recife

2022

Catalogação na fonte
Bibliotecária Nataly Soares Leite Moro, CRB4-1722

C433r Chagas, Amirton Bezerra
A recommender system to support the development of context-aware intelligent transportation systems / Amirton Bezerra Chagas. – 2022.
158 f.: il., fig., tab.

Orientador: Carlos André Guimarães Ferraz.
Tese (Doutorado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2022.
Inclui referências e apêndices.

1. Redes de computadores e Sistemas distribuídos. 2. Sistemas de recomendação. 3. Sensibilidade a contexto. 4. Sistemas de transporte inteligente. 5. Taxonomia. I. Ferraz, Carlos André Guimarães (orientador). II. Título

004.6 CDD (23. ed.) UFPE - CCEN 2022 – 106

Amirton Bezerra Chagas

“A Recommender System to Support the Development of Context-Aware Intelligent Transportation Systems”

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Doutor em Ciência da Computação: Redes de Computadores e Sistemas Distribuídos.

Aprovado em: 08/03/2022.

____
Orientador: Prof. Dr. Carlos André Guimarães Ferraz

BANCA EXAMINADORA

____
Prof. Abel Guilhermino da Silva Filho
Centro de Informática/UFPE

____
Profa. Dra. Patricia Cabral de Azevedo Restelli Tedesco
Centro de Informática/UFPE

____
Prof.Dr. Divanilson Rodrigo de Sousa Campelo
Centro de Informática /UFPE

____
Profa. Dra. Thais Vasconcelos Batista
Departamento de Informática e Matemática Aplicada/UFRN

____
Profa. Dra. Vaninha Vieira dos Santos
Departamento de Ciência da Computação/UFBA

I dedicate this text to my parents, Marli and José Emirton, for the unconditional love, incentive, and support they have given in all moments of my life.

ACKNOWLEDGEMENTS

To my lovely wife Lorena, for enduring this long journey with me since the start, always providing me love, care, attention and motivating me to extract my best.

To all my friends, who didn't let me down on the most stressing moments and provided me incentive and joy during this laborious work.

To E. Favre, J. N. Daniel, J. Buchanan, and C. Yoovidhya, for their hard work on the development of accessible quality products that helped me during this research.

To my advisor Carlos Ferraz, for the great patience, guidance, support, and knowledge provided.

To all my teachers, from preschool to graduate education, on whose shoulders I stood to achieve the completion of work.

ABSTRACT

The development of Context-Aware Intelligent Transportation Systems (ITS) requires a careful analysis to identify which Contextual Elements can contribute to the definition of the application's Context. This activity is complex, especially in the ITS scenario, very vast and with hundreds of possibilities. The objective of this research is to analyze the use of context-awareness in ITS and propose alternatives for organizing this information to allow the creation of tools that contribute to automating part of the task of identifying useful contextual elements for the development of a new application. A literature review of ITS projects served to map the use of contextual elements. 73 projects were mapped, of which 70 were academic and 3 were commercial. Then, a Taxonomy of Contextual Elements Categories was defined, to increase the granularity of the information and facilitate its use in an automated system. The taxonomy has 79 categories in total. A knowledge base was built relating the 73 projects to the taxonomy categories. An experiment showed an increase of 197.5% in the amount of contextual elements correctly chosen when designing an application, when the engineer has knowledge and access to the taxonomy. Using the taxonomy and the knowledge base, we designed a Contextual Element Category Recommender System for ITS. Using an initial subset of Contextual Elements already identified as necessary for a new application, it can recommend categories of Contextual Elements for the subsequent analysis by the application designer. The recommender system validation demonstrated its ability to recommend categories relevant to projects. When using a number $n >= 8$ of similar projects to identify the categories, even limiting the number of recommendations to 15 items, in more than 75% of the time the system recommended categories known to be used for the subset informed as input. The creation of a taxonomy associated with the development of a recommender system using a knowledge base of projects in the ITS area presented the potential to contribute positively to the design and development of applications in this domain, allowing the identification and consequent use of more relevant contextual elements for the project application.

Keywords: recommender systems; context-awareness; intelligent transportation systems; contextual elements; taxonomy.

RESUMO

O desenvolvimento de Sistemas de Transporte Inteligentes (ITS, do inglês Intelligent Transportation Systems) Sensíveis ao Contexto necessita de uma cuidadosa análise para identificar quais Elementos Contextuais podem contribuir na definição do Contexto da aplicação. Esta atividade é complexa, principalmente no cenário de ITS, muito vasto e com centenas de possibilidades. O objetivo desta pesquisa é analisar o uso de sensibilidade ao contexto em ITS e propor alternativas de organização desta informação para permitir a criação de ferramentas que contribuam para automatizar parte da tarefa de identificação dos elementos contextuais úteis para o desenvolvimento de uma nova aplicação. Uma revisão da literatura de projetos de ITS serviu para mapear o uso de elementos contextuais. Foram mapeados 73 projetos, dos quais 70 acadêmicos e 3 comerciais. Com o mapeamento, procedeu-se à definição de uma Taxonomia de Categorias de Elementos Contextuais, para aumentar a granularidade da informação e facilitar seu uso em um sistema automatizado. A taxonomia conta com 79 categorias no total. Uma base de conhecimento foi construída relacionando os 73 projetos às categorias da taxonomia. Um experimento apontou um aumento de 197,5% na quantidade de elementos contextuais corretamente escolhidos ao projetar uma aplicação tendo conhecimento e acesso à taxonomia. A partir da taxonomia e da base de conhecimento, foi projetado um Sistema de Recomendação de Categorias de Elementos Contextuais para ITS, que utilizando um subconjunto inicial de Elementos Contextuais já identificados como necessários para uma nova aplicação, é capaz de recomendar categorias de Elementos Contextuais para a posterior análise do projetista da aplicação. A validação do sistema de recomendação indicou sua capacidade de recomendar categorias que são relevantes aos projetos. Ao utilizar um número $n >= 8$ de projetos similares para identificar as categorias, mesmo limitando a quantidade de recomendações em 15 itens, em mais de 75% das vezes o sistema recomendou categorias sabidamente utilizadas para o subconjunto informado como entrada. A criação de uma taxonomia associada ao desenvolvimento de um sistema de recomendação utilizando uma base de conhecimento de projetos da área de ITS apresentou potencial de contribuir positivamente no projeto e desenvolvimento de aplicações deste domínio, permitindo a identificação e consequente uso de mais elementos contextuais relevantes para a aplicação em projeto.

Palavras-chaves: sistemas de recomendação; sensibilidade a contexto; sistemas de transporte inteligente; elementos contextuais; taxonomia.

LIST OF FIGURES

LIST OF TABLES

LIST OF FRAMES

LIST OF SOURCE CODES

LIST OF ABBREVIATIONS AND ACRONYMS

ADAS	Advanced Driver Assistance System
AI	Artificial Intelligence
APTS	Advanced Public Transportation System
FHWA	Federal Highway Administration
GHS	Globally Harmonized System of Classification and Labelling of Chemicals
GLONASS	Global Navigation Satellite System
GPS	Global Positioning System
GUI	Graphical User Interface
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
ITS	Intelligent Transportation Systems
ITSC	International Conference on Intelligent Transportation Systems
KB	Knowledge Base
k-NN	k-Nearest Neighbor
LIDAR	Light Detection and Ranging
LIVE	Laboratório de Inovação Veicular
LSA	Latent Semantic Analysis
OBD-II	On-Board Diagnostics II
OWL	Web Ontology Language
QoI	Quality of Information
RADAR	Radio Detection and Ranging
RSU	Road-Side Unit
SMC	Simple Matching Coefficient
SUV	Sport Utility Vehicle
V2I	Vehicle-to-Infrastructure

V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-Everything
VANET	Vehicle ad hoc Network
VIN	Vehicle Identification Number

CONTENTS

1 INTRODUCTION

Vehicles are central to the modern way-of-life. Commuting is an important part of everyday life, and having to spend hours in traffic is common in large cities throughout the world. Motorized transportation has evolved from merely mechanical machines to electronically complex devices, full of embedded micro-controllers even in simple vehicles. Smart cities also followed the trend, and now it is possible to control their traffic using software that monitors the current situation of streets and acts according to traffic levels, accidents and any other occurrence that can affect the mobility of a city (FARIA et al., 2017).

In the past, cars were just mechanical machines. The electronic revolution has embedded microcontrollers even in the simplest vehicles. Now, the driving activity is assisted by dozens of onboard computers and millions of lines of code (DIBAEI et al., 2020). This digitalization of vehicles and transportation as a whole is supported by Intelligent Transportation Systems (ITS). This type of system uses the available data from vehicles and Road-Side Unit (RSU), historical databases and other sources to improve a vast number of aspects of transportation, such as vehicle efficiency, city traffic, and route planning. Public transportation systems and other systems more focused on features useful to the Smart Cities domain have also their peculiarities. Occupancy and waiting times, and other features are helpful to smart city planning of their public transportation systems.

Context-awareness is key to the success of this kind of system due to the heavy reliance on data that define the situation of the system actors and the need to react to such situations accordingly. Designing and developing context-aware applications is a non-trivial activity. In particular, the selection/definition of *Contextual Elements*, as some rich context-aware applications may require a good number of them, and the quality of such contextual elements, need to be assessed appropriately. It is known that context-awareness introduces design challenges for software development professionals (HENRICKSEN; INDULSKA, 2006) and that developers need support in enumerating the possible contextual states in their systems as well as in knowing the information that could be used to correctly determine the state of the system (ALEGRE-IBARRA; AUGUSTO; EVANS, 2018).

Also relevant is the fact that, even with all these different concerns to be taken into consideration, it is usual that members of an ITS development team are neither experts in ITS nor in the specific domain of the application (e.g., Advanced Driver Assistance System (ADAS), and teleoperation). That makes it harder for software development professionals to be aware of all the possibilities of using context when creating these applications. Understanding the ITS domain, and thus all the possible contextual elements that can be incorporated to further enriching an application is hard for such a team. A well-performed requirements elicitation process can identify some characteristics that can be mapped onto contextual elements to be used in the system under development. However, it would be hard to identify all the possibly useful contextual elements to the system in a feasible time during the elicitation process.

Several applications have already been designed and developed in the ITS domain, making extensive use of contextual elements to provide useful and rich functionality. Modeling information that can be sensed in or retrieved from a vehicle environment helps to improve the development process, and consequently, the quality of ITS applications. Understanding which contextual element categories could be useful in this environment allows one to create applications adaptable to the current context to offer the best possible user experience.

Applications with few contextual elements (or little contextual data) often provide a poorer user experience. More context-related data processed by the system during its execution could enrich the context information to enable a better user experience (HU et al., 2014). For example, an application that utilizes not only location information (e.g., home, work) but also context information related to activities (e.g., mobility pattern) and environments (e.g., temperature) tends to offer a better service to its users. However, simply increasing the number of contextual elements in an application development does not solve the identified problem. The various contextual elements must be well-articulated, and it is complex to integrate a large number of possible Contextual Elements that may be relevant to the application.

A relevant part of such a complexity is in modeling the context-awareness needs of the application, which implies on identifying the contextual elements that will be used by the application. In this research, the process of defining the contextual elements is improved by the use of a recommender system, that uses as one of its inputs a taxonomy of contextual element categories that was designed based on 73 projects available in the academic literature or that are widely used and are part of the daily life of millions of people. The recommender system proposed must be able to perform the automatic recommendation of specific Contextual Element Categories for a new application under development based on a set of other Contextual Element Categories that have been used in existing ITS applications. Besides, given that requirements engineers must still analyze the recommended categories and that time is often a scarce resource in software development, limiting the number of recommended items to facilitate their analysis and make sure the recommendation's quality is still good enough is an important issue. So we are looking for the best set of recommended Contextual Element Categories that can be analyzed by a requirements engineer in a feasible amount of time and still enable a rich context-aware application. The following scenario illustrates the goals of this research.

1.1 MOTIVATING SCENARIO

This scenario focuses on a software development team responsible for creating a context-aware ITS. Consider an experienced software engineer in charge of designing an ITS application based on a set of requirements received from an also experienced requirements engineer. The requirements are clear and precise, having been elicited using the appropriate techniques, and all the stakeholders of the application were involved in the process. The engineer proceeds to analyze the requirements. While not an expert in the ITS domain, he believes that the requirements seem to contain enough information to allow the adequate system design to be created.

Soon, the software engineer realizes that using context-aware techniques would be an interesting idea in this application. He tries to identify in the requirements which contextual elements should be considered to represent context in this particular system. At this moment, he realizes that while the requirements are quite complete, the information explicitly available in the specification would not be enough to correctly assess the context needed in that application.

The engineer brings a senior developer with prior experience in ITS applications to help identify such aspects. Based on the projects he has previously worked on, the senior developer could find elements related to the ones noticeable in the requirements. Although helpful, the developer's contribution is most likely to be limited by the projects on which he has worked.

All the involved professionals realize that they are lacking structured information on how to model such a context-aware application based on a wider knowledge base, grounded on prior experience of other similar projects on the same domain. A quick search did not identify any work containing all the information they require. They estimate the cost of a research to identify such topics, but it would be very time and resource consuming, not feasible regarding the project's schedule and budget.

Nevertheless, the application is then modeled and developed. While not a bad or poor system, it frequently receives feedback pointing out problems of the application. The feedback usually suggests improvements related to not having considered other contextual elements in the application design.

Fixing the problems and implementing the improvements have proven not to be cheap, as supported by Boehm's curve of software cost of change (BOEHM, 2006). Both problems and opportunities for improvement have been found too late in the development process, what makes changes more expensive and riskier. Because of this, automotive car makers, still regarded as a very conservative and risk-averse industry (SOVACOOL et al., 2019), resist and decide not to make the changes.

A new work that arises after the development of that application provides the information that the engineer needed during its design. Alongside the information regarding a taxonomy of contextual elements, a recommender system is also provided. This system receives an initial set of contextual element categories and promises to deliver suggestions of other contextual element categories usually related to the ones inputted, based on a set of existing and published projects. The final result is an enriched (recommended and larger) set of contextual element categories ordered by their *relevance indexes* to the project.

The engineer, eager to validate both the work and his system, follows the proposed method involving the recommendation of contextual element categories. After analyzing the results, he concludes that the suggestions arising from the method of the new work mostly match the contextual elements, which would avoid the negative feedback received by the application.

This scenario illustrates the context where the proposed work would be an adequate contribution in current development processes, indicating the expected result of the method proposed in this research in the software modeling process.

1.2 PROBLEM STATEMENT

Designing context-aware applications is a challenging task for software development professionals. One of the difficulties introduced by context-awareness is identifying which Contextual Elements are relevant to a particular application, usually a non-trivial job (HENRICKSEN; INDULSKA, 2006; ALEGRE-IBARRA; AUGUSTO; EVANS, 2018) performed by requirements engineers during the requirements elicitation phase. While requirements elicitation is usually a task for specialized requirements engineers, this task is in practice executed by several other software development professionals (FALCAO et al., 2021). While some of the contextual elements are straightforward to be identified, some other ones may require knowledge both on context-awareness and on the specific domain of the application. The vastness of the ITS domain aggravates this problem. Even requirements engineers with previous experience in the domain might need to elicit requirements to an ITS application that is so different from what they have already done that this previous experience can be not so useful in the context of the new application.

Aggregating knowledge from other projects with similar characteristics is not trivial. Even when surveys and other works contain such information, it is not structured in a way that eases the analysis. Thus, designing the context-aware application to its full potential soon in the development process is not always feasible due to the lack of structured and easy-to-use information. The lack of organized and accessible information regarding context-awareness and contextual elements usage in previous ITS applications impacts the ability of designing new applications of this domain that use the full potential that context-awareness could provide to them. Thus, identifying contextual elements that can be used in an application is currently limited to the information that the professional responsible for the requirements elicitation already knows or had direct contact with during the elicitation process for this application.

Furthermore, this same lack of structured information inhibits the creation of automated tools that can reduce the engineers workload by performing some of the tasks during the design, such as identifying potential contextual elements that can be useful in the new application. An automated tool that is able to provide such suggestions could both reduce the workload of the requirements engineer and also allow for the serendipitous discovery of contextual elements that could not be perceived given the data available to such a professional.

The chosen domain is an interesting option since context-awareness is salient to ITS applications. Safety and security issues, which are most commonly related to the domain, clearly benefit from the use of context-aware techniques. ITS, however, also deal with issues such as infotainment (characteristics associated with providing information and entertainment to drivers and passengers alike), optimization of resource usage such as fuel, and other concerns which are also positively affected by context-related information, such as traffic improvement and public transportation management.

1.2.1 Research Question

Based on the presented concepts, it is possible to raise a question (*main research question - RQ*):

How to assist software development professionals on the discovery of useful contextual elements for ITS applications?

The aforementioned research question leads to the following secondary questions, also related to the problem:

- *RQ1*: Which contextual elements are useful to ITS projects?
- *RQ2*: How automatic the discovery of contextual elements useful to an ITS project can be?
- *RQ3*: Is it viable to generalize the approach to other domains?

1.3 OBJECTIVES

The main objective of this research is to investigate how the process of eliciting the usage of contextual elements in an ITS application can be improved. Organizing and using information available of such usage from projects in the literature and in the market is important to achieve this objective. This organized information can than be used directly by software engineers as a source of knowledge during the design of new ITS applications, or indirectly through tools that use the organized knowledge to provide services such a recommender engine, which given a previously known subset of contextual elements, maximize the use of other related contextual elements useful to the application.

1.3.1 Specific Objectives

The specific objectives of this research are:

1. Identify which contextual elements have already been used in academic or real-world projects of the ITS domain;
2. Identify approaches to aggregate information about usage of contextual elements in existing context-aware applications;
3. Propose a recommender engine that can be used by software development professionals to discover useful contextual elements to their ITS applications.

1.4 CONTRIBUTIONS

The main contributions and the respective secondary contributions of this work are:

- A Taxonomy for ITS Contextual Elements
  - The definition of categories of Contextual Elements used in the ITS domain
  - An evaluation of the usefulness of this Taxonomy during the design of an ITS
- A knowledge base of context-awareness usage in ITS applications
- A process to provide automatic recommendations of Contextual Element Categories to ITS
  - A process to calculate a relevance index to order the recommendations according to their potential relevance;
  - An evaluation process to measure the effectiveness of a contextual element categories recommender.

1.5 THESIS STRUCTURE

The remaining of this thesis is organized in the following chapters:

- Chapter 2 contains all the relevant background that was investigated during this research, with an analysis of the areas of Context-Awareness, ITS, Recommender Systems and Similarity Measurement, all used as base to this work. In the end of this chapter, an analysis of Related Works is also performed.
- Chapter 3 explains the Taxonomy of Contextual Element Categories that is proposed in this work. The chapter contains subsections to explain the methodology used to perform the research that led to the creation and validation of the taxonomy. It also contains a detailed explanation of each of the categories that compose the taxonomy, a section explaining the evolution of the taxonomy and how it was validated, and a section with a discussion of the proposed taxonomy.
- Chapter 4 is focused on the part of this research that, based on the proposed taxonomy, designed and created a recommender system to help engineers on designing new ITS

applications. It also contains a section to describe the methodology used to design the recommender system and an explanation of some artifacts used during the creation of the recommender system. A section explaining the inner working of the system is also presented, then the evaluation process that was devised and followed to check the results provided by the recommender. The results of this validation are shown and explained in section 4.4, and a section focused on a discussion of the recommender system is also available.

- Finally, Chapter 5 contains the final remarks about this research, including a section with threats to validity and another section containing the future works that can be derived from this research.

2 BACKGROUND AND RELATED WORK

Our research combines different research topics to achieve ITS goal. This chapter discusses and provides a background on Computational Context, Intelligent Transportation Systems and Context-Aware Vehicle Applications, Recommender systems, and Similarity Measuring. It ends with a discussion on related works that categorize and model context in applications of the ITS domain.

2.1 CONTEXT-AWARENESS

Dey (2001) defined context as “any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves,” and such context is usually related to the location, identity and state of particular entities or groups of entities. Dey’s definition indicate that systems can improve their efficiency and user experience by clever usage of contextual information.

Zimmermann, Lorenz e Oppermann (2007) enhance DEY’s definition and state that “Elements for the description of this context information fall into five categories: individuality, activity, location, time, and relations.” Furthermore, they affirm that there are links between some of these categories, such as the connection of time and location as primary generators of relations between entities.

Vieira et al. (2007) provide a complementary definition of a Contextual Element as “any piece of information that enables an agent to characterize entities in a domain.” This definition is fundamental for this research since, based on a categorization of such elements, it serves as the base for recommending which information could be missing in a given project.

Bauer e Novotny (2017) use the term *Intelligent Systems* as a synonym to Context-Aware systems. The authors mention that the general concept of systems which adapt according to context is also referred to by other terms in the literature, such as intelligent, adaptive, situated and ambient.

An extensive review of definitions for context awareness was done by Temdee e Prasad (2018). Their work, through an analysis of the most commonly used definitions, concluded that no agreement on a general definition of context exists. However, they were able to establish their common points. Temdee e Prasad (2018) identified that there is a common agreement that context is:

- Any information used to characterize the situation of entities;
- Able to give meaning to other things, either by itself or in combination with other context information;
- Generally used to describe the operational aspects of context-awareness, instead of the inherent properties that define context.

Another point of convergence in the works of Dey (2001), Vieira et al. (2007), Abowd e Mynatt (2000) and Zimmermann, Lorenz e Oppermann (2007) is the need for classification of useful data for context reasoning. The five Ws of Abowd e Mynatt (2000) (Who, What, Where, When, and Why) are a basic and general way of categorizing contextual information. Dey (2001) and Zimmermann, Lorenz e Oppermann (2007) *dimensions* are a step further in modeling context regardless of domain. Vieira, Tedesco e Salgado (2009) define an important concept to distinguish *Contextual elements* from *Context*. *Contextual elements* are unique pieces of information that together can define the *Context*, which is "the set of instantiated *Contextual Elements* that are needed to support the task at hand".

Alegre, Augusto e Clark (2016) observe that categorization of context is also difficult, and different approaches to provide taxonomies for context have been attempted, each with their particularities. However, it is noticed also that the authors of each categorization recognize that it is not possible to define a single scheme able to abide to the requirements for all kinds of context-aware applications.

Adding context-aware capabilities to a software system is not straightforward. As already mentioned, Henricksen e Indulska (2006) describe some of the challenges of designing context-aware systems. Indeed, since the start of researches involving context awareness, investigating its impacts in software development practices is a popular line of research. Hong, Suh e Kim (2009) surveyed the research work on the context awareness field and found that 5.5% of the articles surveyed were related to development guidelines, representing the eighth most common type of research in the context awareness domain, from a total of 26 categories.

Context-awareness is a mature field of study, with more than two decades since the work of (DEY, 2001). It continues to be relevant and widely used in applications of different domains (CHAVHAN et al., 2021; MENSAH; MWAKAPESA, 2022). In domains related to smart cities, such as ITS, context-awareness is a commonplace technique, recognized for allowing the creation of useful applications. As such, there is a large number of context-aware applications when considering smart cities environments (ROCHA et al., 2022).

2.2 INTELLIGENT TRANSPORTATION SYSTEMS

The Institute of Electrical and Electronics Engineers (IEEE) organized the first International Conference on Intelligent Transportation Systems (ITSC) in 1997 (IEEE, 1997), but academic research and industry development of more efficient transportation systems can be traced back to the 1930's (FIGUEIREDO et al., 2001). In the mid-1990s, the fundamentals of ITS were strong enough to allow the research and development communities to adopt unified methods, processes, and terminology to projects in the topic.

Based on the definitions given by Figueiredo et al. (2001) and Guerrero-Ibanez, Zeadally e Contreras-Castillo (2015), it is possible to establish that an Intelligent Transportation System is the application of communication, information, and electronics to minimize pollutant emissions, vehicular wear and time spent on commuting while maximizing fuel efficiency, road usage, and safety. Guerrero-Ibanez, Zeadally e Contreras-Castillo (2015) state that emerging technologies, such as the trend of connected vehicles, cloud computing and the Internet of Things (IoT), will shape the future development of ITS. With the hindsight of seven years having passed since this prediction, it is proving true given the current developments and projects that are emerging in the area, some of which are further mentioned in this work in the next sections.

2.2.1 Context-Awareness in ITS

Use of context-awareness in ITS applications is very common. Indeed, research by Golestan et al. (2016) and Sepulcre e Gozalvez (2018) conclude that cars need to be aware of the current situation so ITS applications are able to achieve their goals. Several examples of projects can be found in the literature, such as the works of Silva, Borges e Vieira (2018), Aguirre et al. (2017), Ali, Muhammad e Khan (2020), Chavhan et al. (2021), and Vieira, Caldas e Salgado (2011).

A large number of ITS application projects already make use of context-awareness. Vahdat-Nejad et al. (2016) present an extensive survey with the state-of-the-art of context-aware vehicular network applications, categorizing them according to different criteria, such as service type, context type, context gathering methods, environment type, system architecture, and others. Different vehicular applications try to solve a wide range of problems in this domain, ranging from security issues, traffic congestion avoidance, environment protection, information, entertainment and driving comfort. The same study shows that applications can focus on urban, freeways or both environments. While some applications were still in the design phase, there are already plenty of context-aware vehicular applications deployed. Overall, the work of Vahdat-Nejad et al. (2016) help in supporting the point that applications in this domain are very diverse but share some common characteristics.

Silva, Borges e Vieira (2018) presents a context-aware system that attempts to assess whether the moment is appropriate for presenting notifications to the driver based on data from smartphone sensors. Based on gyroscope and Global Positioning System (GPS) data, the system identifies situations such as whether the vehicle is stopped, moving in constant speed and direction, moving in a curve or in the process of changing lanes. Rules were defined to decide whether the moment is opportune of innoportune for presenting a notification to the driver.

The context model of this application uses five contextual elements: Acceleration, Speed, Angular Velocity, Distance Traveled and the Driver's Driving Activity. Values for these elements are obtained from the APIs offered in Android to access the sensors' data.

One differentiating aspect in the work of Silva, Borges e Vieira (2018) is that it can be considered a subsystem to many other ITS applications. Any project where there is the requirement of notifying drivers with not-urgent information can base this feature implementation on the results of Silva, Borges e Vieira (2018). Indeed, a similar feature is avaliable while navigating on Waze, where interruptions on the map screen, mostly for ads, happen when the car is on-route but stopped in traffic lights, or stuck in traffic.

Aguirre et al. (2017) propose an approach to use wireless sensor networks in a context aware application to support urban train transportation, with environment monitoring of users' interactions. The application monitors environmental information such as temperature, humidity, and CO and CO_2 levels. Nodes deployed to stations know in which station they are (there is an identifier for each station), and sense the trains that are into the station at a given moment. The application also can be modified to allow the identification of the passenger's location and provide content-based advertising. The system allows for users to subscribe to receive updated information of the monitored values.

Vieira, Caldas e Salgado (2011) presents a prototype of *Your City on Time*, an Advanced Public Transportation System (APTS) to estimate bus arrival time at bus stations based on contextual information, using contextual elements from the bus, the roads and the bus stations.

From the bus station, its static location is used. The location, speed and average speed of the bus are also collected, alongside which line it is currently operating, and elements related to the relationship between a bus and the line's stations, such as the last station it has passed by, and the distance and time to arrive at the next station. Information regarding the stretches of road relevant to the bus journey are also used, such as the traffic level and the average speed of vehicles passing by that stretch. All this information is then used to infer the time required for the next bus of a line to arrive in each station.

The application is mostly autonomous, without requiring user interactions to be used. It is expected that users only consume the information displayed in screens or panels regarding their desired bus lines.

It is interesting to observe that currently many cities worldwide use systems that have the same goal and use similar approaches to the application proposed by Vieira, Caldas e Salgado (2011). The Moovit application, available in more than 3400 cities in over 100 countries, is an example of a system that has similar goals (SANTOS; NIKOLAEV, 2021) and supposedly uses similar techniques.

The previous examples of ITS help to make clear that the domain is vast and diverse, not limited just to vehicular applications such as ADAS, which is a common misconception about the area. While vehicular application that aid the driver or the vehicle itself are integral parts of the ITS domain, other applications more in line with the concept of smart cities are also a big part of the systems in the domain. Additional examples of ITS projects are:

- Dikaiakos et al. (2007): investigation of how to develop time-aware services to provide useful information to drivers, such as traffic conditions and nearby road services.
- Seredynski, Arnould e Khadraoui (2013): research of using vehicular networks for interconnecting traffic lights and vehicles, allowing the signal lights to adapt to real-time traffic conditions as well as providing feedback to drivers with information about the appropriate speed to drive such that the vehicle can reach the next stop lights when they are green.
- Reis et al. (2009): research that found evidence that the efficiency of individual sen-

sors providing their data to vehicles can be improved by aggregating on a contextual-processing device. Their work aims to create this device to integrate location, weather, traffic lights, and camera data to further process and generate new useful information for the transit scenario.

A successful real-world Intelligent Transportation System application is Waze. The application's website <http://www.waze.com> mentions that it is used by more than 140 millions of drivers and carpoolers worldwide, Waze is currently almost a synonym with getting directions and navigating. The application features a map and allows drivers to input an origin and destination. Waze then calculates a route based on several information such as the road conditions, speed limits, current speed on each stretch of road reported by instances of Waze running in other users' devices, road blocks and a multitude of other information useful to finding good routes for users.

The system also aids drivers while they are navigating the map, by providing, aside from the route navigation per se, information on traffic jams, law enforcement, speed cameras and potholes that were reported by other users. More recently, a carpooling feature was added to the system, where a driver can share its ride with other users (in the role of pedestrian and passengers). Users needing a carpool input their route and the system looks for drivers who are on-route and can, with a minimal detour, provide the carpool.

Waze is a very good example of an ITS which reached widespread use, and is probably the most used by common people on their day-to-day activities. Its benefits are perceived by any user who have already needed to drive in an unknown city, or live in urban areas with heavy traffic. Also, it is heavily context-aware, adapting constantly to the surrounding environment the user (and also of any point in the user's current route).

Recent research such as those by Swarnamugi e Chinnaiyan (2020), Chavhan et al. (2021), and Dzemydienė e Burinskienė (2021) show that context-awareness in ITS is a relevant and effective approach to enrich this kind of application. However, it is common to find research similar to the one performed by Zheng et al. (2016), where the usage of Big Data on Social Transportation is reviewed, mentioning several topics clearly related to context-awareness, but without even mentioning the word "context". Waze is another example of this phenomenon, since no explicit mention to context-awareness is available in their website or merchandising material, even with the strong use of context-awareness in the application. A careful researcher in this area must be aware of this fact, because projects related to ITS generally make use of computational context.

2.3 RECOMMENDER SYSTEMS

Recommender systems are a useful type of tool which, based on previous knowledge about users and domains, are able to create suggestions fit for a given situation. They allow the decision-making process by humans to become faster and more accurate by filtering the options that at a first glance could be too vast to allow a good choice by the user (QUIJANO-SÁNCHEZ et al., 2020; KARIMI; JANNACH; JUGOVAC, 2018).

According to Schrage (2020), Recommender Systems are able to help users on deciding what they could do, explore relevant options, compare the available options and discover new options and opportunities that they might not have been able to find out by themselves. Thus, common uses for Recommender systems are generally related to given a large amount of data, perform computations to identify useful patterns that a human could otherwise not been able to find in a timely fashion.

Aggarwal (2016) states that Recommender Systems have four operational and technical goals. They must provide **relevant** results to users, which is straightforward to understand and is essentially the primary goal of any recommender system. In many systems, recommendations usually should also be **novel**, to avoid showing only popular items or items that the user already knew. **Serendipity** is another goal of a Recommender System, in the sense that some of the recommendations must be unexpected, or not obvious. Such recommendations will help offering new relevant possibilities to the users which they could have a hard time identifying by themselves, and are important differentiators of recommendation systems. Finally, one last goal that recommender systems will usually have is providing **diverse** recommendations. Recommender systems in online sales, arguably the most common domain where this type of system is used by regular pepole, are very incentivized to provide diverse results, which can in their turn provide a better conversion of a recommendation into a sale, instead of providing several similar results.

Recommender systems are being used for over three decades, and are increasingly needed in a world where the massive amount of data available makes impossible for humans to analyse and navigate through such data (SCHRAGE, 2020). As already mentioned, e-commerce is one of the most dominant areas to recommender systems, and is also the area where most people had their first contact with such systems. Schrage (2020) argues over the importance of recommender systems to the success of the Amazon.com store, mentioning how the low cost and scalable personalization provided by this kind of system helps on keeping users satisfied and buying more, which converts into the success of the store.

Clearly, their use is not limited to the online sales domain. Other well known cases are discussed by Aggarwal (2016) and Schrage (2020), such as the Netflix movie recommender system, which enables quality recommendations of their huge catalog to their users and has been a key differentiator of the company since it was based on the mail DVD-rental business model. YouTube and other streaming services also make use of recommender systems to increase viewer satisfaction and time consuming the content. Still on the streaming area, Spotify recommender systems generate personalized playlists based on large data the company has regarding the songs, artists, user preferences and user history.

Many other use scenarios for recommendation systems exist: The Google News personalization system, where relevant and diverse news are recommended to users based on their history and context. Social networks (now commonplace) friend recommendations, and online dating matching sites, all use recommendation systems of the most various types to achieve their business goals (AGGARWAL, 2016; SCHRAGE, 2020).

It is possible to categorize Recommender Systems in several different manners according to the approach used to generate the recommendations. There are several different alternatives to perform such categorization. Some examples are: Sridevi, Rao e Rao (2016), with the proposal of twelve approaches to the development of Recommender Systems; and Sheshasaayee e Muniyandi (2020), using eight different categories of approaches. Based on the definitions given by Ricci, Rokach e Shapira (2011) and also used by Aggarwal (2016), it is possible to identify three classes that are more relevant to this research and will be further detailed in this section: Content-based, Collaborative filtering, and knowledge-based systems.

2.3.1 Content-based recommender systems

Information about the items that can be recommended can be valuable to the process of recommendation. For instance, in a music recommendation system, the song's genre, release date and the artist's country of origin can be important attributes to suggest similar items, when the user listening history is analysed for these attributes. Thus, content-based recommenders assume that users will be satisfied by receiving recommendations of items that have similar properties to those that they have consumed and liked previously (SCHRAGE, 2020).

Content-based recommender systems usually also depend on information regarding the user's previous actions, preferences, and profile data to be able to recommend other content similar to these past interactions and previously known information (KARIMI; JANNACH; JUGOVAC, 2018). So, it is possible to affirm that content-based systems use two sources of data: the item attributes data and the user profile (AGGARWAL, 2016).

Moreover, they require modeling the available content that is subject to be recommended in such a way that it is possible to compare them with the user's profile. The item's attributes information can usually be modeled as a matrix, relating each item to the value of the features that they have for each attribute (SCHRAGE, 2020). The comparison is commonly based on similarity measurement functions, which are further described in Section 2.4. There are also other ways to achieve similar results by using ontologies that model the domain of the recommender system (QUIJANO-SÁNCHEZ et al., 2020).

New items are not an issue for content-based recommender systems. Since they use the item attributes, it is not necessary that a set of users rate an item (either directly, by assigning a score or liking, or indirectly, by buying, viewing or otherwise consuming the item in some way) so it can be recommended to other users (AGGARWAL, 2016). So, if a new item is registered in the system, it can instantly be recommended to users which profile and history matches the attributes used in the recommender system. Thus, content-based recommender systems don't have the item cold start problem (FALK, 2019). Aggarwal (2016) reinforces that the content-based recommender systems do not use the rating that an user provides to influence the recommendations to any other user.

Another advantage of the content-based approach is its transparency (GEMMIS et al., 2015), that allows clear explanations of the reasons for a recommendation. It is possible to indicate which attributes were considered, their values and their similarity to the user profile with ease.

One of the limitations of content-based recommender systems is dealing with new users, which still do not have enough profile data collected to allow providing useful recommendations. As such, while this type of recommender system is immune to the item cold start problem, it is succeptible to the user cold start problem. Another known limitation is overspecialization, where the system always recommends items so similar that users may lose interest in the output of the system (GEMMIS et al., 2015). A similar related problem is that, by not using ratings provided by other users, the recommendations can be obvious, lacking the serendipity that is expected from this type of system, due to being usually strongly related to the past consumption history of each particular user (AGGARWAL, 2016). Finally, according to the scenario where the content-base recommender system will be applied, identifying the most useful attributes of the items that must be used to obtain an accurate recommendation for a specific user can be challenging (SCHRAGE, 2020).

Aggarwal (2016) speculates that the recommender engine used in Pandora (Pandora Media Inc., 2022) uses techniques from the content-based approach, using attributes such as features of the tracks, and expecting some initial user feedback to create a recommendation model, overcoming this way the user side of the cold-start problem that is present on this approach. Lin et al. (2017) points out that Airbnb property ratings are mostly very positive, with over 95% rated 4.5 or 5 stars, which could hinder the possibility of using Collaborative filtering. Thus, they propose a content-based filtering scheme using reviewer data and property features. While it is not the real implementation of Airbnb's currently used Recommender system, Lin et al. (2017) observation on the difficulties of using collaborative filtering with Airbnb data suggests that the company might use some kind of content-based filtering, either alone or in an hybrid approach. Thanh-Tai, Nguyen e Thai-Nghe (2016) also mention that CiteSeer uses content-based filtering to recommend interesting papers in its library.

2.3.2 Collaborative filtering recommender systems

Item-related information is useful and can provide good results when used in recommender systems, as mentioned in the previous section. However, they tend to generate results that are not so diverse, with a potential to become unsurprising and boring to users. This weakness of content-based filtering can be remedied by the use of collaborative filtering.

Collaborative filtering recommender systems utilize the user's past actions as input to compare with ratings provided by other users with similar actions (QUIJANO-SÁNCHEZ et al., 2020). These systems are based on the assumption that users with similar behaviors would share the preference for the same items. Instead of focusing on the items properties, the focus is on how other users interacted with the items, and on identifying users with similar behaviors, so items well-rated by one of these users can be recommended to the other (AGGARWAL, 2016). It is important to emphasize that a collaborative filtering-based recommender system is able to provide relevant results even without any specific knowledge of the items, relying only on the patterns of usage or ratings of the system's user on the items (KOREN; BELL, 2015).

Interestingly, according to (SCHRAGE, 2020), Collaborative filtering has its roots in a computerized system, however manual, developed at the Xerox PARC, where users manually tagged documents and performed actions on them. The system provided querying capabilities not only for document data, but also for the tags and reactions that were applied on them. The need for manual input and a direct action of users for both tagging as well as retrieving data (also known as pull-active collaborative filtering) limited the success of the initial application. This was improved later in other projects, with more automation on generating tags and ratings, and on implementing push-based techniques to offer more direct access to the system helped the model on gaining traction. Nowadays, such automation is made possible by techniques such as k-Nearest Neighbor (k-NN), Bayesian Networks, Correlation metrics and Neural Networks, which are commonly used in this kind of system (SRIDEVI; RAO; RAO, 2016; SHESHASAAYEE; MUNIYANDI, 2020; QUIJANO-SÁNCHEZ et al., 2020).

Systems applying this principle are vastly used to provide recommendations for content such as music and movies, and are considered to have a good accuracy. They are not so heavily affected by the new-user problem, and the over-specialization issue also is not usually considered a problem in such systems (AGGARWAL, 2016).

However, they present some problems, such as item cold start (SRIDEVI; RAO; RAO, 2016) or susceptibility to becoming a target of malicious users (LI et al., 2016). The item cold start issue arises because in this type of filtering, new items are problematic due to the absence of ratings and interactions of users with them, which without external inputs would make it less probable for the system to recommend such new items. Falk (2019) presents a good overview of the Cold Start problem, including its subtypes, real examples and potential solutions.

Aldayel e Benhidour (2019) states that collaborative filtering approaches are popular in e-commerce systems. Indeed, it is known that Amazon.com uses item-based collaborative filtering recommender systems since 1998, with enormous business success attributed to it (SMITH; LINDEN, 2017). It is also known that, in the video streaming domain, both YouTube and Netflix also use collaborative filtering to provide useful reccomendations to their users (SMITH; LINDEN, 2017).

2.3.3 Knowledge-based recommender system

A knowledge-based recommender system depends on some kind of explicit representation of the knowledge of its domain, allowing the recommendation of items that fulfill the user's desires (FELFERNIG; TEPPAN; GULA, 2007). In contrast with content-based and collaborative filtering techniques, this one requires previous work on understanding and modeling the domain.

However, this extra work in the system design comes with the advantage of not suffering from issues concerning cold start (FELFERNIG et al., 2011) or misleading biases caused by malicious users (SRIDEVI; RAO; RAO, 2016).

Ricci, Rokach e Shapira (2011) point that it is common that knowledge-based recommender systems attempt to estimate the utility of their recommendations based on heuristics that hypothesize how relevant a recommended item will be to the user. Such heuristic, or utility function, essentially represents the probability that a user will consider the recommended item useful. This function is designed and provided to the recommender system in advance, like another piece of external knowledge. Domain-specific knowledge is often used to design a utility function shaped to rank potential recommendation items in a particular system (AGGARWAL, 2016).

Aggarwal (2016) highlights that knowledge-based recommender systems may generate large lists of results. Leaving to the user the task of filtering the results on their own can be burdensome. Constraints can be applied to decrease the size of the result list. These constraints can have default values and also be configured by users during their use of the recommender system.

Regarding the techniques used to create knowledge-based recommender systems, it is possible to divide them into two major categories: Constraint-based systems, where rules are defined and the system follows these rules to determine the items that should be recommended; and Case-based systems, which use similarity metrics to identify which items are more likely to be useful to the user based on its input (FELFERNIG et al., 2011; SRIDEVI; RAO; RAO, 2016).

In Constraint-based recommender systems, the previous knowledge is represented by an extensive mapping of the rules that guide how the items are interrelated. These rules are then implemented and followed by the recommender engine. Thus, it is necessary a previous analysis of how the relations and the overall recommending process. While in some domains this is possible, in other domains such analysis might be too hard, or the data is too entangled to allow an assertive answer.

Regarding Case-based systems, it is possible to affirm that:

> A Case-based recommender is a Knowledge-based system that exploites a 'search and reuse' approach. The search is performed on the catalogue of items (to be suggested), and the reuse of retrieved items could be implemented in different ways, from a simple display of the retrieved items to a more complex adaptation of the items to fit to the peculiar preferences of the user" (RICCI et al., 2006).

In case-based recommenders, the similarity of the items in the knowledge base in relation to the case provided by the user is used by the recommender engine to provide recommendations. This shifts the focus from understanding how the items are interrelated to the discovery and selection of items to compose a reasonable-sized knowledge base that will represent the previous cases.

Furthermore, it is interesting to recognize that Recommender Systems are the evolution of Expert Systems (TINTAREV; MASTHOFF, 2011). The rule-based technique commonly used in expert systems is also worth being observed due to its competence to help decision-making processes emulating human experts' behavior (ENGIN et al., 2014). Similar to knowledge-based recommender systems, rule-based expert systems also rely on a previously collected knowledge base and are usually focused on a single domain. An inference engine follows the rules in the knowledge base to generate the output of these expert systems (ENGIN et al., 2014).

Knowledge-based recommender systems have been used in several domains, such as tourism planning (RICCI et al., 2006; JANNACH; ZANKER; FUCHS, 2009), financial services (FELFERNIG et al., 2007) and effort estimation in software development (PEISCHL et al., 2009). Aside from being used as the single recommender approach in a system, it is also common to find usage of knowledge-based recommender systems in an hybrid approach with collaborative filtering or content-based filtering, to overcome the cold start issue (BOBADILLA et al., 2012; BAHRAMIAN; ABBASPOUR; CLARAMUNT, 2017; ARNAOUTAKI et al., 2019).

2.3.4 Comparison of Recommender System approaches

The previous subsections provide a glimpse into the vastness options that are available in the design and development of recommender systems. While other approaches do exist, such as ontology-based recommender systems (usually categorized as a subtype of the knowledge-based approaches) and even context-aware based recommender systems (ADOMAVICIUS; TUZHILIN, 2015), the presented approaches are the most important to understand the choices made on this work.

Also, it is possible to observe that each of the approaches have its strengths and weaknesses. Understanding these characteristics of each approach is essential choose an appropriate design to a new recommender system. Frame 1 contains a comparison of the approaches on dimensions that we considered relevant to this research.

It is possible to observe that for the problem that our recommender system intends to solve, Frame 1 – Comparison of Recommender System approaches

Approach	**Item cold start**	**User cold start**	**Benefits from previous knowledge**	**Works with few items**	**Allow increasing the base data**	**User provides input**
Content-based	Unnafected	Affected	Partially	No	Yes, automatically	Indirectly
Collaborative	Affected	Unnafected	Partially	Partially	Yes, automatically	Indirectly
Knowledge-based Constraint-based	Unnafected	Unnafected	Yes, rules	Yes	No	Directly
Knowledge-based Case-based	Unnafected	Unnafected	Yes, items	Yes	Yes, manually	Directly

Source: The author (2022)

using a collaborative filtering approach is non-advisable. The cold start would definitively be a problem in our scenario, and the number of users and items would not be enough to a system of this kind work properly.

Constraint-based approaches, while not affected by the cold start issue, would require the identification of rules to create the recommender. This is not feasible in our scenario, since no relevant visible pattern emerges from an observation of the data.

Content-based approaches could be used, since with some tagging effort on the categories of the taxonomy, we could devise a way to use this technique. However, the number of categories is low and it is considerably static, since it will only change in the event of updates to the taxonomy. This nature of our data makes using this approach also not an interesting option.

The Case-based approach is particularly fit to the needs of this research. Being a type of knowledge-based recommender, it has no issues with new users and cold start. One of its major problems is the lack of learning components, which can render them less useful over time. This is not, however, a problem to the application of this technique in the context of this research, and is also surpassable in the future if required, by adding a learning component to the system. Since case-based approaches are based on similarity measurement, the next section will provide an overview of the available alternatives to calculate the similarity between the user input and the items in the knowledge base.

2.4 SIMILARITY MEASURING

As previously mentioned, Case-Based recommender systems rely on the similarity of a case with items of a knowledge base to work properly. Thus, it is important to use an objective approach to calculate and measure such a similarity. Several measurements of similarity exist (VIJAYMEENA; KAVITHA, 2016) and could be used in such a situation, with different degrees of accuracy on the similarity results.

The simplest of them is the Simple Matching Coefficient (SMC). Equation 2.2 shows how to calculate the similarity using SMC, where the similarity value between two projects could be given by the total number of matching contextual element categories used (given in Equation 2.1, where BP is the set of Contextual Elements Categories in the *Base Project* and CP is the set of Contextual Element Categories in the *Compared Project*) divided by the total number of categories available (T).

$$MCE(BP, CP) = |BP \cap CP| \tag{2.1}$$

$$SMC(BP, CP, T) = \frac{MCE(BP, CP)}{T} \tag{2.2}$$

SMC, however, is not very useful in our scenario because it counts equally as a match of the presence of a contextual element category in both projects as well as its absence in both of them. Indeed, it is not particularly well suited to situations where there is asymmetry in the real value carried by each possible outcome. In our scenario, the information of the presence of the contextual element category carries much more value than the information that a project does not use any element of that category.

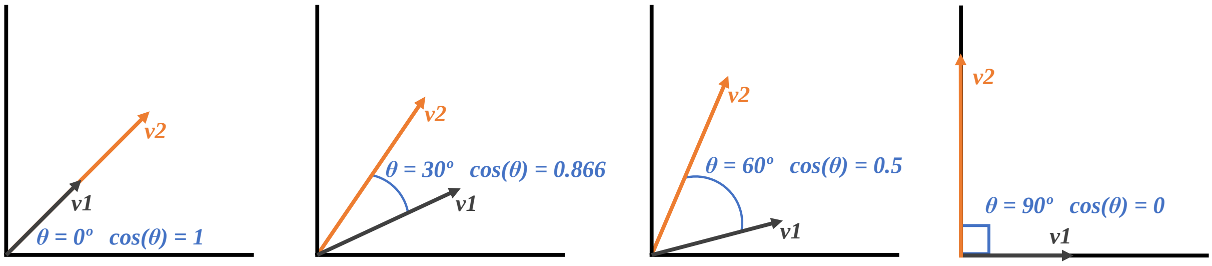
The *Jaccard Coefficient* can be used when such an asymmetry exists. Equation 2.3 explains how this is achieved: instead of using the total contextual element categories as the denominator, when applying this measurement we only consider the total number of categories which are present in at least one of the projects.

$$JcC(BP, CP) = \frac{MCE(BP, CP)}{|BP \cup CP|} \tag{2.3}$$

Although the Jaccard Coefficient fits well for our current data, other approaches are more efficient when the data matrix is sparse, as currently is the case with our data. Also, in its pure form it is limited to binary variables. While the data used in the projects matrix has been binary, it could later expand to a continuous value indicating, for instance, the importance of that particular contextual element category in a specific project, or to a discrete value such as the number of elements of that category used in the project.

Another measurement approach, which is recommended when sparse data is in use (HAN; KAMBER; PEI, 2012) and works correctly with non-binary values, is the *cosine similarity*. *Cosine similarity* measurement is based on the cosine of the angle between two vectors. When applied to this research data, each vector is one of the analysed projects, and each possible contextual element category is a dimension of this vector. The presence of an element of a category in a project sets the value of that dimension to 1, while its absence keeps the value as 0. Thus, these vectors represent the project in terms of the contextual element categories used, and the orientation of one of these vectors will be closer to the one of projects using similar categories – the closer, the more similar (see Figure 1).

Figure 1 – Cosine similarity illustrated



Source: The author (2022)

Calculating the similarity between projects, therefore, means calculating the cosine of the angle between their respective vectors ($v1$ and $v2$), as in Equation 2.4 and illustrated in Figure 1, since the smaller the angle between them, the closer to 1 the cosine will be, i.e., the more similar they are.

$$CS(BP, CP) = \frac{BP \cdot CP}{|BP| \times |CP|} \tag{2.4}$$

Cosine similarity is commonly used in Recommender Systems (ZIARANI; RAVANMEHR, 2021) to find objects in a knowledge base which are similar to another given object. Such objects can be the behavior log of users, groups of products, or any other set of related items. Indeed, Amatriain et al. (2011) states that cosine similarity is usually accepted as one of the best choices of similarity measurement for Recommender Systems. Another field where the use of cosine similarity is widespread is text matching, where the cosine similarity between texts is an alternative to identify a different text with the same overall subject. In this research, it is a proper choice since it behaves correctly with multidimensional matrices of binary elements. Also, due to the aforementioned way it calculates the similarity value, the absence of a relation does not influence in the result, which is also something desirable in our sparse matrix of data.

2.5 RELATED WORKS

The related works of this research can be divided into two groups: Those related to Taxonomy and Modeling, and those that deal with automatic recommendations in software design. The following subsections discuss the works found for each of these groups.

2.5.1 Categorization and Modeling

We performed a literature review looking for other taxonomies and models focused on contextual elements for generic vehicular applications. We used Google Scholar, searching for the queries *"context-aware" categorization*;*"context-aware" classification* and *"context-aware" modeling*. A filter for works published since 2010 was applied. We ignored articles where the title indicated that they were off-topic. For the others, we analyzed the abstract, and for those where the abstract indicated that it would be possibly related to this work, we proceeded with the reading of the full article to confirm or deny this suspect. We did not find any work that summarizes such information in this domain. However, some alternatives specialized in sub-domains of context-aware vehicular applications were found.

Vahdat-Nejad et al. (2016) provide a categorization of context types into four groups. Each of these groups contains context information that shares similar characteristics. One group is local context, for information that describes local entities, such as the vehicle where the system is running or its driver. Another group is the external context, which describes the same information but is related to other vehicles or drivers in the scenario. Vahdat-Nejad et al. (2016) define other two groups: General-related to transportation, for information that is directly connected to the context of transportation but is not related to a vehicle or its driver (such as parking information); and General-unrelated to transportation, for any other information relevant to an ITS but not related to transportation, such as weather.

The coarse granularity of the categorization defined by Vahdat-Nejad et al. (2016) is interesting for their objective of identifying types of projects that usually need each kind of information. However, for the aim of helping software professionals to understand better the possible uses of context-awareness in their ITS applications, it is too coarse. We would need a categorization in finer granularity based on contextual elements.

On the opposite direction of the coarse-grained categorization provided by Vahdat-Nejad et al. (2016), we have found the ontology defined by Klotz et al. (2018). The ontology provides a very fine-grained definition of signals available in the vehicle. It can go as low as to identify specific doors or seats in the vehicle, for instance. Our research does not need such a low level of detail, and yet the ontology is limited to vehicle information, but still Klotz et al. (2018) is a source of inspiration of our model.

Kannan, Thangavelu e Kalivaradhan (2010) used an ontology based on Web Ontology Language (OWL) to model context regarding vehicle safety. It defines contextual elements and their relationships for safety-driven context-aware vehicular applications. Xiong, Dixit e Waller (2016) also propose an ontology focused on contextual elements related to the driving activity, and most of their use scenarios are around ADAS, sometimes extrapolating the ontology's use to self-driving vehicles. Like other already discussed ontologies, this one focuses on a particular sub-domain of ITS (driving). Driving, for instance, is only part of what the taxonomy proposed in our research covers.

Haupt e Liggesmeyer (2021) focuses on the categorization of context for Autonomous Vehicle Safety. The work starts oon the classification of context based on its relevance to their environment, into either the relevant or irrelevant categories. Then, in the relevant category, they sub-categorize the model further, until reaching the level of the contextual information. While the design decisions and the scope of the classification performed by Haupt e Liggesmeyer (2021) is different from our research, the overall idea of a hierarchical classification of context information is present and central on both works.

2.5.2 Automatic Recommendation during Software Design

Mougouei e Powers (2021) have proposed a system to automatically select software requirements based on their value and their relation to other ones. Even though our research objective is quite different from the objectives of Mougouei e Powers (2021), it is an interesting work based on an expert system to support decisions that will guide the software design and development. The work uses Integer Programming and Fuzzy Graphs to model and identify the values of the dependencies and generate an optimal subset of the requirements based on the limitations given as input. The importance of dependencies between requirements is also an interesting point that has a parallel in our research, as the results given by the recommender system proposed in our work are entirely dependent upon the relationship between contextual element categories.

Designing a method to improve the design of context-aware systems is one of the objectives of Engelenburg, Janssen e Klievink (2019). Some of the steps in their proposed method allow the system designer to identify the contextual elements that the application needs and which components are required to sense or act given these elements. Their proposed method is a collection of steps to be manually followed by a system designer. Initially, this characteristic suggests a contrast with our approach since we provide a recommender system to identify potential contextual elements. However, a closer look allows us to verify that our proposed recommender system could be used as an improvement to their steps where contextual elements identification is performed, avoiding missing important contextual elements due to the factors mentioned in Chapter 1.

Falcão et al. (2021) investigates the effect of exposing software development professionals to a data-driven context model on the identification of context-aware-related features, including the identification of contextual elements, during the phase of elicitation of context-aware functionalities. They perform an experiment that is similar in the concept and execution to the blind experiment performed in our research and described in Section 3.5.1. Their results, after exposing the experiment group to the context model used in their research, also has expressive results, with a 139% increase on the number of contextual elements used to describe the requirements when compared to the requirements elicited by their control group. It is a work that follows a different approach than ours on the instrument used to improve the elicitation of context-aware requirements, but is similar in the overall objective of allowing better designed context-aware applications by providing more tools and information to the professionals performing the requirements elicitation.

Using expert systems to help identify potentially incomplete requirements and checking for their consistency has indeed been done for some time already. Sinha e Popken (1996) proposed a system that performs such checks in the domain of Air Force Weapons Systems. Although the research is not recent, the problems the authors point to as the possible reasons for requirements' incompleteness are still valid and correlated to the reasons for selecting an incomplete set of contextual elements to a context-aware ITS application. The reasons are: the engineer might forget to elicit all possible requirements; in case of multiple system designers involved, one designer might believe that another one is already documenting a requirement; system complexity or engineer inexperience might make it hard for listing all potential requirements; the designer might be designing a system without full knowledge of the properties desired for the system.

Sinha e Popken (1996) design and validate an expert system to verify the completeness and consistency of the requirements using a knowledge base with the domain knowledge from several years of previous systems designed in the same domain. A correlation matrix is used to find potential relations among requirements and the rationales that justify their selection.

The recommender system proposed in our research shares some of the same objectives and characteristics of that system, but working on a very different domain. All the previously mentioned reasons for requirements incompleteness apply to the ITS application domain. One of our objectives is also easing the process of system design, attempting to use an expert system to help engineers identify potentially missed characteristics that might be valid to the systems they are designing. Different from their modeling, we do not use an object-oriented approach to model our knowledge base. Our domain allows the use of a relational knowledge base, which simplifies the creation of our correlation matrix. Our recommender system also differs in the user interaction approach, as it generates the output with all the recommendations and their relevance index values to the engineer, so it is possible to evaluate the fitness of each recommendation to the given system.

Dumitru et al. (2011) propose a system to recommend features to a product based on an initial set of known features (called a *profile*) and a collection of product descriptions. Their system obtains the descriptions through mining large public repositories (e.g., Softpedia) containing product descriptions. They propose a clustering algorithm and use association rule mining and standard k-NN approach to recommend additional features given the initial profile. While the reasoning approach used by their recommender system and our proposal are very different, the general objective of both works is similar: Given a small and incomplete set of known characteristics of a new system under design, recommend new characteristics based on other previously created similar systems. In their case, the characteristic is the feature, while in our case, it is the contextual element categories. It is important to notice that their approach requires using a large dataset as a knowledge base, which is possible for general software features. Our proposal, in the specific domain of ITS applications, is more restricted in these terms. To the best of our knowledge, as previously mentioned, there is no available collection of data where contextual element categories used in previous ITS applications could be mined, which justifies our different approach to achieve a similar goal.

Tang et al. (2019) reviewed plans to implement smart city strategies in 60 cities in Africa, Asia, Europe, North America, and Oceania. One of the analyses of this work consisted of defining 25 categories of smart cities-related projects and measuring the occurrence frequency of projects from each of those categories in the plans of the analyzed cities. Then, the probability (p) of co-occurrence of pairs of project categories was calculated as $p > 0.72$. It was defined based on the values of the top quartile to represent the more likely co-occurrence pairs. There are interesting characteristics to be learned from Tang et al. (2019). While they considered that using conditional probability was fit to their problem, we are not able to assert the same in the case of our data, since conditional probability requires that no causal relationship exists among the compared scenarios. We consider that using cosine similarity to compare the projects is an adequate approach, but the inspiration of using a conditional probability-based matrix and comparing its results with the results obtained from the use of cosine similarity could be an interesting future work.

2.5.3 Comparison of Related Works

The two main contributions of this research are from two distinct areas. The related works are very specific to each of these areas, and as such, we decided to separate the analysis and comparison of these works into two separate sections to provide a better understanding of the works and how they compare to each of the contributions in this work.

2.5.3.1 Classification and Categorization Related Works

Frame 2 summarizes the characteristics of the related works regarding the categorization and modeling of contextual elements, allowing a better understanding of where our proposal fits in this scenario.

It is possible to observe that our taxonomy proposal differs from the related works for providing a grained taxonomy of contextual elements specific to the domain of ITS, which encompasses any application of this area, not only ADAS or VANET, for instance.

Another observation that can be made from Frame 2 is that most of the works related to the classification of context elements use the ontological approach. While these works provide correct justifications for such choice, mostly based on the power of representation of knowledge that is inherent to ontologies, this approach would not be a good fit for the objectives of our

Frame 2 – Comparison of related works - Taxonomy

Related Work	Domain	Technique	Granularity
Vahdat-Nejad et al. (2016)	Vehicle ad hoc Network (VANET)	Hierarchy of Dimensions	Very Coarse
Klotz et al. (2018)	Vehicle Signals	Ontology	Fine-grained
Kannan, Thangavelu e Kalivaradhan (2010)	Vehicular Safety	Ontology	Grained
Xiong, Dixit e Waller (2016)	ADAS	Ontology	Grained
Haupt e Liggesmeyer (2021)	Vehicular Safety	Categorization	Coarse
Our Proposal	ITS	Taxonomy	Grained

Source: The author (2022)

research. A more simple approach is required, so it can be understood without the requirement of having knowledge about ontologies or any other richer, however more complex, instrument. More information on this design choice is provided in Chapter 3.

2.5.3.2 Related Works on Assisting the discovery of Contextual Elements for an application

Frame 3 summarizes the characteristics of the related works regarding the discovery of contextual elements that can be used in an application. This includes approaches such as the recommendation of contextual elements, allowing a better understanding of where our proposal fits in this scenario. Also in this table, we included the comparison of related works that deal with the evaluation for consistency or completeness of requirements or contextual elements in an application.

None of the related works proposes a recommender for contextual element categories. Further discussion on the reasons for the design choices we took on the proposed Recommender system can be read on Chapter 4.

Frame 3 – Comparison of related works - Aid to the requirements elicitation of context-aware applications

Related Work	Technique	Advantages	Drawbacks
Sinha e Popken (1996)	Knowledge-based Expert System	Control over all the projects that are part of their knowledge base; Correlation matrix is straightforward to use	Complex modeling of the knowledge base; Lack of ordering on the results
Dumitru et al. (2011)	Text Mining and Diffusive Clustering	Can use data from partially-structured sources; General domain allows for a larger previous knowledge base; Works with small input	Approach requires large dataset to work
Tang et al. (2019)	Conditional Probability	Approach can work with small knowledge base; Not affected by the cold-start problem	No casual relationship must exist among the scenarios in the knowledge base; Knowledge base manually built
Engelenburg, Janssen e Klievink (2019)	Manual process to evaluate contextual elements	Part of a complete method for designing context-aware applications; Can be followed and validated by multiple persons	Manual process; Time consuming; Error-prone
Falcão et al. (2021)	Data-driven context model	Data-driven approaches are rising in popularity on improving requirements engineering; Do not require experience on context-awareness	Manual process; Effort to create the context model
Mougouei e Powers (2021)	Integer Programming and Fuzzy Graphs	Allows the selection based on limitations of the system under design; Considers dependency between requirements	Large dataset required
Our Proposal	Case-based Recommender System using Similarity measurement	Similarity matrix allows use of sparse data in the knowledge base and input; Provides a relevance index to order the recommendations; Adequate results with small knowledge base; Works with small input; Not affected by the cold-start problem	Knowledge base manually built

Source: The author (2022)

3 CET-ITS: CONTEXTUAL ELEMENT TAXONOMY FOR INTELLIGENT TRANSPORTATION SYSTEMS

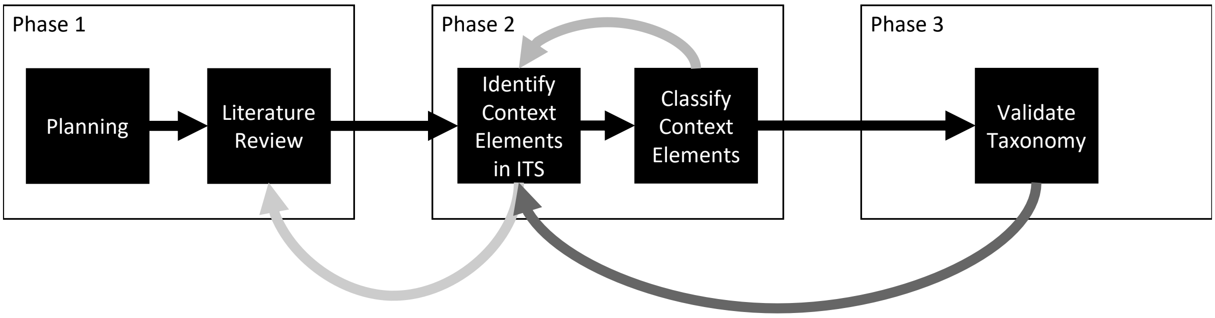
We consider the definition of *Contextual Elements* given by Vieira, Tedesco e Salgado (2009) as a starting point to our proposal. Understanding the concept of a Contextual Element as a single piece of information that can carry enough value to be processed and infer the current context is a central factor in the definition of our model. In this thesis, the term *Category* (or *Contextual Element Category*) is used to define a group of Contextual Elements that are related to one another by some characteristic. The characteristics that define each of the categories will be further discussed in this section. Our initial proposal presents a hierarchical model where each subcategory has only one parent category. A Contextual Element however can be classified to more than one category, albeit this is not common. Most Contextual Elements are classified only into a single Contextual Element Category. While we understand the limitations of favoring such a restricted model over a richer ontology-based model, where categories can relate to each other, we also realize that such a simple model has advantages.

Understanding a hierarchical model is straightforward because the relationships are simpler and more linear. While a comprehensive ontology formalizes the relationships between different elements in the modeled world, such relationships can visually pollute the graphical representations of the model (LOHMANN et al., 2016) and may not be relevant to most users. A hierarchical classification (categorization) focuses on establishing only the strongest links between its elements, freeing readers to define less relevant relationships that might be useful to their scenarios. The concept of representing useful contextual information using this type of model has already been defined for other domains such as Web Navigation, Medical Monitoring applications, and health software defects (VILLEGAS et al., 2011; MITCHELL et al., 2011; RAJARAM et al., 2019).

3.1 METHODOLOGY

The methodology to create the taxonomy is similar to the one defined by Papatheocharous et al. (2018) and is depicted in Figure 2. The taxonomy was built in three main phases. In the first phase, we identified the problem through a literature review on Intelligent Transport Systems and Context-Aware Systems, in general, to understand their development process and identify how a taxonomy could help improve this process.

Figure 2 – Modeling the proposed taxonomy



Source: The author (2022)

The second phase was the design of the taxonomy, starting with the identification of contextual elements used in the ITS applications found in the projects reviewed in the literature. In this stage, a careful reading of the articles describing each project was performed. Each identified contextual element was written down. While the relationship between a project and a contextual element has not been registered during this research, it could have been useful and would have saved a considerable amount of time in further analysis.

After concluding the analysis, the next step consists of the elements being checked in the taxonomy, compared according to the similarity of their characteristics, to verify for the existence of a valid category where that contextual element could be classified. This classification was performed by the author of this work, with validations and improvement suggestions arising from the evaluation processes that were performed on the taxonomy. The criteria to classify a context element to an existing category was:

- The contextual element is really of the type of the supra-category of the analysed category?
- The contextual element is similar to other contextual elements that were classified in this category?
- The semantics of the contextual element agrees with the semantics of the category?

When this analysis failed for every potential category where the contextual element under analysis could fit, we proceeded to the creation of a new category. In such case, the criteria for the positioning of the new category in the existing hierarchy was as follows:

- Choose the adequate supra-category for the contextual element;

- Find the bottom-most middle-level category where the criteria to classify a context element to an existing category was valid
- Create a new category, child of the category found in the previous step;
- Review the known contextual elements to check whether they would be more adequate in this new category.

The third phase was the validation of the taxonomy, which is detailed in Section 3.5 of this work. Similar to the process followed by Papatheocharous et al. (2018), there were several iterations between the second and third phases, which served to improve the taxonomy design, that is, to evolve the taxonomy (Section 3.5). In some cases, these iterations also needed to go back to the literature review, intending to find articles about new ITS projects that could help identify more contextual elements and possibly new contextual element categories.

3.2 LITERATURE REVIEW

This section provides an overview of how the literature review was performed to guide us on finding relevant ITS projects.

The literature review was to find articles in scientific journals and conference proceedings that describe existing Intelligent Transport Systems. There was no specific filter regarding context awareness, as systems commonly have context-aware features without explicitly mentioning them. The tool initially used to search for these articles was Google Scholar. As this tool provides results from a large number of sources, some of which are not entirely reliable, we carried out reputation checks on articles that meet our criteria but have not been published by renowned publishers such as IEEE, Elsevier, or Springer.

Our criteria included search terms such as *"Intelligent Transportation System"*, *"Intelligent Transportation Systems"*, *"Vehicular Application"* and *"Smart Cities" "transportation system"*. Search results where the title makes clear that the work is not about a specific ITS project or a survey of ITS projects were not analysed. The next step was filtering by an analysis of the article's abstract, which ruled out many of the results also. Articles that went through this sieve were subject to more detailed analysis. Many projects were considered not eligible to use in this work because their use of context was not clear, or the contextual elements used was not assertively defined in their articles. Since our work depends on the accuracy of the relation of the elements, we opted for not including the projects from articles where it was not possible to be sure of which contextual elements were used.

Articles were found that describe systems developed since 1998, and even if the older projects are outdated, the set of contextual elements used by them can still be useful and represent valid information to assess the categories defined in the taxonomy.

Five surveys were identified (VAHDAT-NEJAD et al., 2016; BARAS et al., 2018; GOMES et al., 2020; KHEKARE; SAKHARE, 2012; SOYTURK et al., 2016). The subject of the surveys was varied but connected to vehicular applications and context awareness. The projects cited in those surveys were also analyzed to check which ones were appropriate to this research's objectives, since the subjects of the surveys always encompassed our domain, but were also broader than it.

The literature review done for the taxonomy resulted in 70 projects collected from 68 papers. Two of the papers describe two projects each: Bifulco, Amitrano e Tregua (2014) reports two cases of ITS usage, one in Singapore and the other in Amsterdam; and David et al. (2013) presents four smart city-related systems, two of them (Loading Zone Management and Communicating Bus Stop) being useful to compose our knowledge base. 41 of the papers were identified through the previously mentioned surveys: 19 papers were collected from (VAHDAT-NEJAD et al., 2016), 7 from (BARAS et al., 2018), 9 from (GOMES et al., 2020), 2 from (KHEKARE; SAKHARE, 2012), and 4 from (SOYTURK et al., 2016). Some of the papers appear in more than one survey, so the numbers cited mention only the first occurrence of the paper in a survey. Furthermore, three commercial ITS projects were used during the validation of the taxonomy: Waze, Uber and Moovit.

3.3 OVERALL DESIGN OF THE PROPOSED TAXONOMY

During the second phase, the definitions of context and context awareness were reviewed to identify guidelines on how to categorize context information. Provided that a very important goal of this taxonomy is that it can be easily read and navigated, defining top-level categories was crucial in the design process of the taxonomy.

Our proposed taxonomy model (Figure 3) is based on the four basic context types defined by Dey, Abowd e Salber (2001): Identity, Location, Time, and Activity (or status). Other models include other top-level categories, such as:

- The ontology for context representation in groupware proposed by Vieira, Tedesco e Salgado (2005) (e.g., Interaction, Organizational, and Physical);
- The categories proposed by Kaltz, Ziegler e Lohmann (2005) for web applications (e.g., User&Role, Process&Talk, Location, Device, and Time);
- The classification proposed by Vahdat-Nejad et al. (2016) for contextual information in the transport domain (e.g., Local, External, General-related to transportation and General-unrelated to transportation);

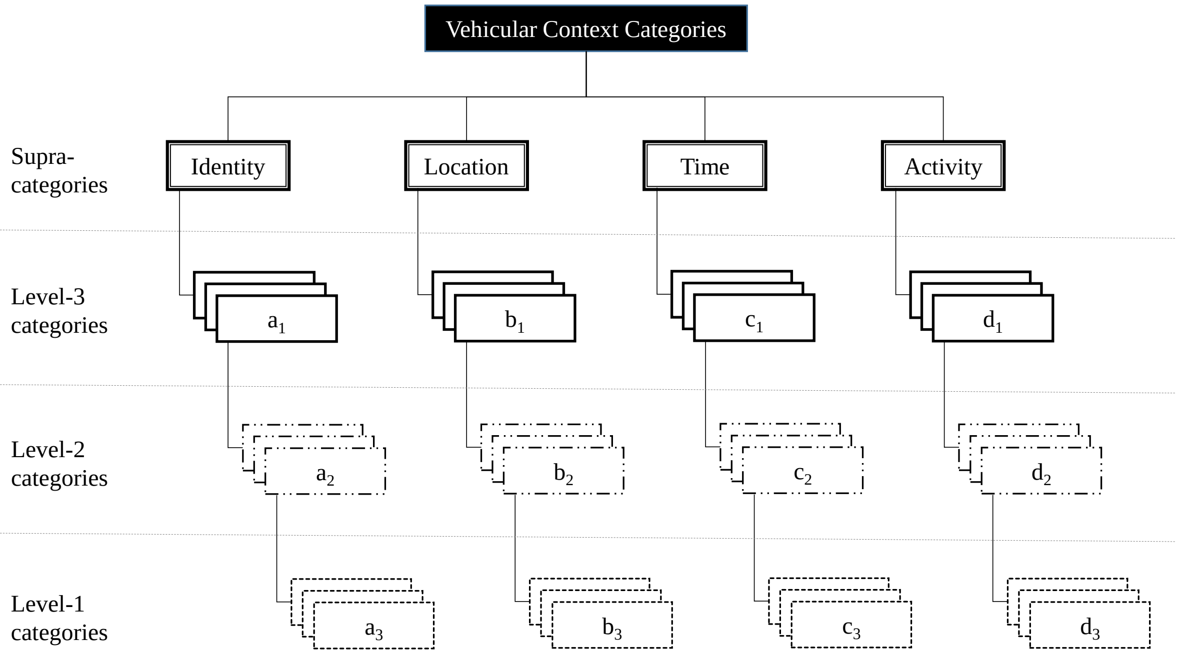
However, the four categories defined by Dey, Abowd e Salber (2001) are still always present either as a distinct or as part of another category. The taxonomy will be presented in parts according to the four basic context types. These four basic context types are called *supra-categories* in this work, as they are the top-most contextual element categories in our taxonomy.

After defining the supra-categories, a design decision also had to be made regarding the depth of the taxonomy. Initially, we had not given a limit to it. However, after the process of reviewing the ITS projects found during our research and identifying the contextual element categories and subcategories that should be part of the proposed taxonomy, we have found no need to go beyond four levels (including the supra-categories).

The subcategories of the supra-categories will be in levels 3 to 1 in Figure 3. These subcategories are present in the taxonomy to further detail and group similar contextual elements. Having three levels under the supra-categories was not a design decision of the taxonomy. Instead, it was a result of the observation on its several iterations that we did not need to go deeper than 3 sublevels of categories. The Time supra-categories only needed to be detailed to the Level-3, while Identity and Activity go all way down to Level-1. Another interesting observation on the taxonomy and its levels come from the fact that most of the Level-3 categories of the Identity and Activity supra-categories are *Entities* in the context of an application. The categories in Level-2 and Level-1 in these cases are groups of contextual elements that characterize those entities.

Entities can be repeated in other points of the taxonomy. The position they appear on the taxonomy is relevant, because a category for the "Driver" entity in the "Identity" supra-category will have subcategories or contextual elements related to the identification and static information of a driver. Another category for the same "Driver" entity, but in the "Activity" supra-category will have subcategories or contextual elements related to the actions and other

Figure 3 – Proposed taxonomy model



Source: The author (2022)

dynamic information of the same entity. A subcategory for the "Driver" entity will also exist in the "Activity/Surroundings/Traffic Participants" category, and in this case, its goal is to be a group of the contextual elements related and relevant to a Driver when it is on the surroundings of the vehicle.

One question that may arise is why focus on defining *Contextual Element Categories* rather than directly proposing the use of basic *Contextual Elements*. The reason lies on the granularity of the information. Using the basic Contextual Elements would lead to using very fine-grained information, with the possible values in the order of hundreds (e.g., the "geographic coordinate" category leads to at least three basic Contextual Elements – latitude, longitude, altitude), while the categories would be in the order of tens. The basic Contextual Elements would be too application-specific, and it would increase the taxonomy complexity because of the greater number of depth levels. Since one of the objectives of the taxonomy is to support the creation of a recommender system (that is further detailed in the Chapter 4), finding relevant relationships among projects (which are also in the order of tens), using too fine-grained information would not be feasible, thus, our choice for defining the taxonomy.

While this 4-level depth is valid for this taxonomy defined for the ITS domain, we cannot assert that this would apply to other domains. However, this depth that we found helpful for ITS allowed managing the taxonomy's complexity and keeping it human-readable and understandable, and it could be used as a goal or soft limit when defining such taxonomies for other domains.

It is important to emphasize that the methodology that guided the design of the taxonomy and the definition of its categories and subcategories is based on the evidence of the use of the contextual elements on projects of the domain. When this use is confirmed by the analysis of a project, the contextual element is analyzed to check whether it fits any of the categories currently in the taxonomy. Only when no current category is adequate to hold a newly found contextual element, we do decide on the addition of a new category to the taxonomy. While this process may lead to categories that are not very detailed, such is the case of the Time supra-category, it ensures that the categories represent the knowledge obtained from the projects that were analyzed in this work.

Further details regarding the projects that were considered in this research and their usage of the contextual element categories can be found in Section 3.5.3, where the most frequently used categories are detailed, as well as the number of categories of each supra-category used in each project, the average number of categories used in articles grouped by year, and the number of articles published in each year.

Following the methodology for modeling the proposed taxonomy, as seen in Figure 2, a second step in phase 2 was the classification of contextual elements, so *Contextual Element Categories* in the ITS domain could be defined.

3.4 CONTEXTUAL ELEMENT CATEGORIES

The contextual element categories represented in the model contain information that can define the context for many different actors in Intelligent Transportation Systems. When the term Vehicle is used, it encompasses both motorized vehicles such as cars, buses, motorbikes, or trucks, and also human-powered vehicles such as bicycles. When contextual element categories using terms like "Driver" are used, we expect the reader to be able to extrapolate the term to "the entity in control of the vehicle", such as a pilot for motorbikes, the cyclist for bicycles, or software for an autonomous vehicle, except for the Person/Driver Id category, where the Driver must be understood only as a human entity. This exception is due to, during the process of defining the categories, we have not found any project which used any information that could uniquely identify the software entity in charge of controlling a specific instance of a vehicle.

Frame 4 shows examples of *contextual elements* relating some of the categories that

Frame 4 – Examples of Contextual Elements in some of the categories. See a comprehensive list in Appendix C

Supra-category	Category	Contextual Element
Activity	Vehicle Movement	Speed
		Acceleration
		Direction
	Cargo	Cargo Temperature
		Cargo Movement
	Driver Status	Tiredness
		Mood
	Driving Task	Hands position
		Eyes direction
	Driver Infotainment	Is Radio On?
Identity	Vehicle Id	Vehicle Identification Number
		License Plate
	Driver Id	Driver's License
		Social Security Number
	Profile	Name
		Age
Location	Symbolic Coordinate	Cellular Base Station ID
		WiFi Networks SSIDs
	Distance Traveled	Total distance traveled by the vehicle
		Distance traveled in current journey
		Distance traveled since last rest stop
Time	Travel Time	Time since travel start
		Time since last rest stop
	Date and Time	Local Time/Date
		Day of the Week

Source: The author (2022)

were defined in the taxonomy, grouped by their respective supra-categories. A longer list with contextual elements for all the 70 categories can be found in Appendix C.

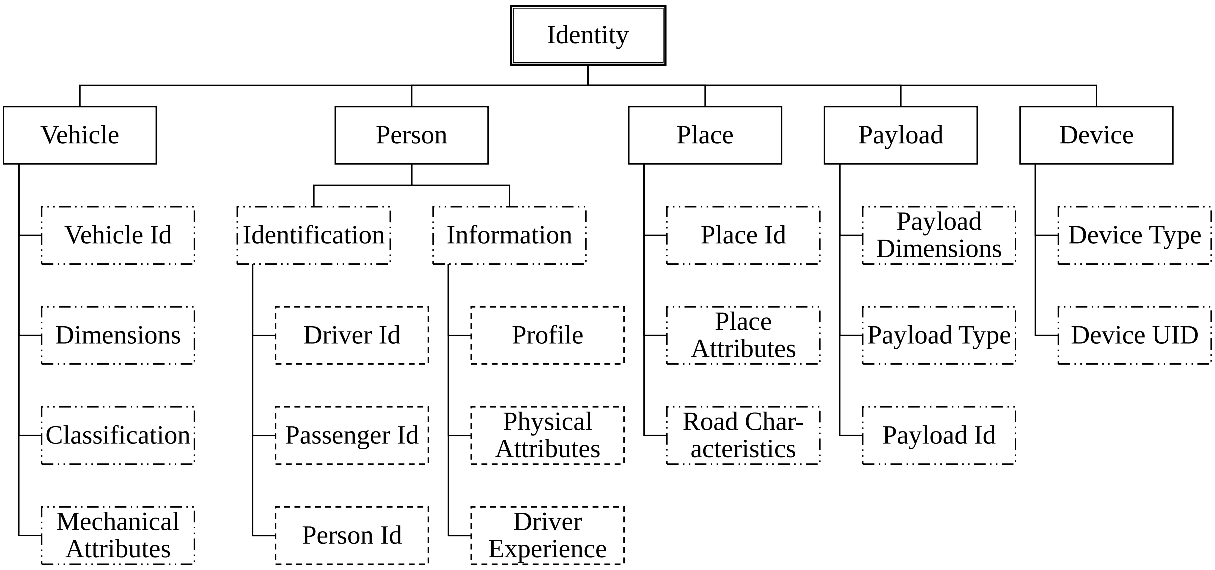
The process for defining the supra-categories has already been described in the previous section. For defining their subcategories, we initially proceeded with the research for relevant ITS projects. Then, we analyzed which Contextual Elements those projects use and, from the supra-categories, tried to classify or categorize the elements that we had already identified, or create a new one. After this initial identification, we reviewed the overall links of the taxonomy, and with this hindsight, rearranged some of the contextual element categories. This iteration was done several times, with improvements coming from the evaluations performed on the proposed taxonomy, as described in section 3.5.3. Also in section 3.5.3, there will be a list of the projects used to help in defining the taxonomy.

The following subsections describe the categories and subcategories that compose this taxonomy. To ease the identification of the level of the categories, the following scheme was applied: Level-3 categories are styled in **bold**, Level-2 categories are styled in *italic* and Level-1 categories are styled both in ***bold and italic***.

3.4.1 Identity

The Identity supra-category is depicted in Figure 4 and consists of information that helps to identify the main elements in the scenario, as well as to characterize them with their immutable attributes. It can be further sub-categorized by defining the main stakeholders in a vehicular application: the vehicle itself, the driver, passengers, and the cargo being transported. Furthermore, the identity of places relevant to the context of the application, and the identity of other devices relevant to the environment where the application is inserted are also important, and the taxonomy contains subcategories for the elements related to these two areas too.

Figure 4 – Identity subcategories of the proposed taxonomy



Source: The author (2022)

Vehicle identity is the first contextual element category we will describe. It has information whose purpose is to portray the vehicle, so that contextual applications can have enough data to uniquely identify and reason about the vehicle's characteristics. Vehicle identity information can be specialized into four categories:

- *Identification*: composed of contextual elements like Vehicle Identification Number (VIN), License Plate, national registration numbers, or any other information that can uniquely identify the vehicle. The vehicle's brand and model are also information that fits in this category.

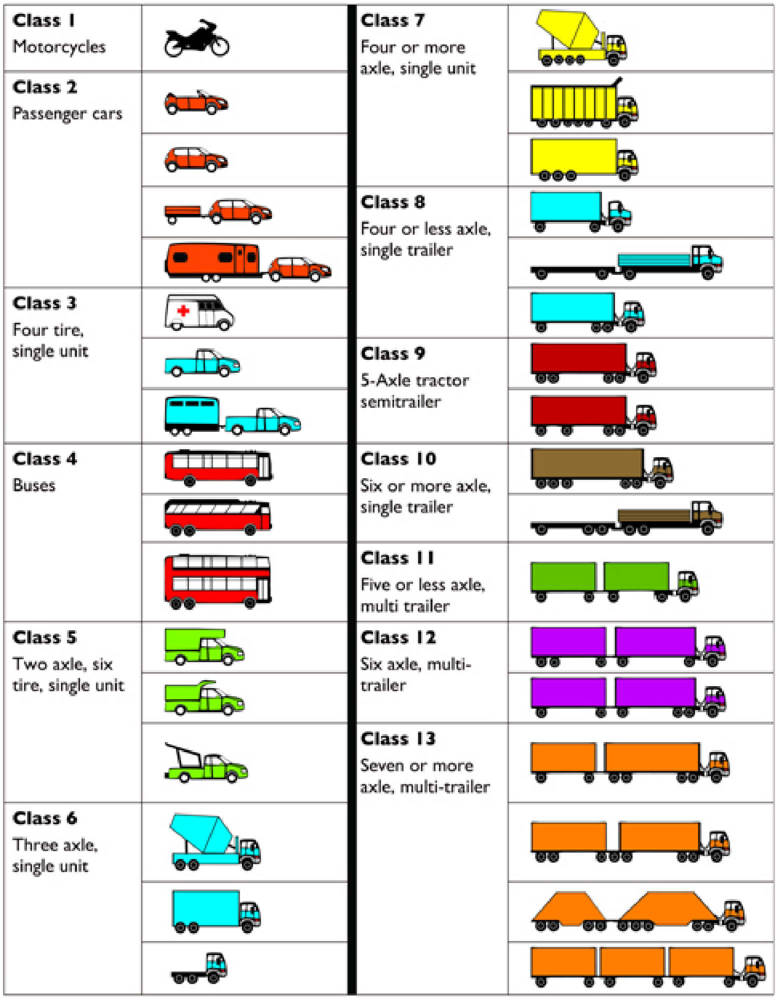
- *Classification*: information about vehicle categorization, such as its type (Hatchback, Sedan, Sport Utility Vehicle (SUV), etc.). While initially this seems to be not a contextual element category but a single contextual element, vehicle classification is complex, and many alternative schemes exist, grouping vehicles according to different attributes. Governments and their agencies can define categorization schemes for vehicles, such as the US Federal Highway Administration (FHWA)'s 13-Category Scheme (Federal Highway Administration, 2016), shown in Figure 5. Businesses that deal with vehicles, such as car rental and car insurance companies will usually have their own classification models for the types of vehicles they work with.

- *Mechanical attributes*: Contextual elements that carry invariable features of car mechanics. Engine displacement (cylinder volume, usually expressed in liters or cubic centimeters or inches – e.g., a 2.0 liter engine), suspension and gear characteristics, and any other information that helps to define the vehicle based on its mechanics.

- *Dimensions*: Information about the size of the vehicle. Length, height, and width, as well as weight specs (unloaded and maximum weights, for instance), are contextual elements in this category. It comprises also the dimensions and weights of possible trailers towed by the vehicle. It is important to emphasize that the vehicle's current weight does not fit in this category, due to its high mutability.

Contextual elements in the vehicle identity category should be immutable, or at least stable enough to rarely change (such as license plates or registration numbers, depending on the jurisdiction). This allows them to be known in advance, requiring limited integration with sensors. Values for these contextual elements can be hard-coded in vehicle's onboard computers, inputted by users on the first use of applications, or fetched from online services covering vehicle identification.

Figure 5 – FHWA Vehicle 13-Category Scheme



Source: (Federal Highway Administration, 2016)

Person identity is equivalent to the former category, but containing elements that define people related to the application, such as the driver, passengers, or pedestrians. This category is divided into the identification and information subcategories.

The *Identification* contextual element category holds three subcategories: ***Driver Id***, ***Passenger Id*** and ***Person Id***. Mostly, the same information, such as a Driver's License or any official identification document, can be part of any of these subcategories. What defines to which of them it is part in a specific system is the semantics: The driver's license, when used to identify the driver, is part of the Driver Id. However, when it is used to uniquely identify a pedestrian, would be part of the Person Id, since that person is not a driver in the context of the application.

Elements from either of these contextual element categories can come from knowledge, physical, and possession characteristics, as defined in terms of authentication factors. Names and official document numbers are examples of knowledge identification factors. These elements will generally be informed by users during setup, sign-up, or log-in to applications. Given that other driver identification elements are present, they could be fetched from local or online databases.

Biometric data are physical identification contextual elements that are part of the Driver Id, Passenger Id, or Person Id categories, according to the entity that they represent. Considering that facial recognition technology is currently mature enough even to differentiate identical twins (LEYVAND et al., 2011), we can use user pictures or 3D mapping data as physical identification of the driver.

Finally, possession-based identification contextual elements can consist of tokens, cards (both contact and contactless), or any other hardware (e.g. smartphone) that can provide data to identify the Driver, Passenger, or any other Person involved with the system, and is supposed to be in their exclusive custody. Devices like these are already in use for security-related applications, such as anti-theft systems and an increasing trend in remote keyless entry systems. This kind of information can be obtained from card readers, but most commonly from wireless sensors to avoid the hassle of fitting keys, cards, or other devices in specific places. When wireless technology is used, drivers can just carry the identification hardware in their pockets and the system will still be able to retrieve the required information.

As already mentioned, most of the types of identification can be part of the Driver, Passenger, or Person Id categories. However, there are some unique elements of Driver Id, such as a professional driver's registration number (within a company, for example), or of Passenger Id, such as a train ticket number.

The *Information* contextual element category contains identity elements that cannot be used to uniquely identify a user. This information is static, or at least it should not vary frequently. Its three subcategories are Profile, Physical Attributes, and Driver Experience.

- ***Profile***: Elements representing general characteristics of persons involved in the application are part of the ***Profile*** contextual element category. Information such as the person's name, address, phone number, birthday, or social media links is part of this category. Infotainment-related preferences, such as music style or preferred radio station are also part of the profile. It can be obtained via manual input, but some of the information can be collected through the use of web services, or even inferred from the continued use of some data. It is important to remember that user profile information can be subject to privacy laws and must be protected accordingly.

- ***Physical attributes***: Applications designed to help improve ergonomics must have access to information such as height, weight, and other more specific physical characteristics. Accessibility-based applications also need to gather information regarding the person's physical abilities. Applications with features of safety and emergency-handling situations can be improved by having access to elements of this contextual element category. Acquiring contextual elements in this category is possible via user input and sensors.

- ***Driver experience***: Another category of contextual elements that can be used by applications. The word "experience" can be considered here encompassing both the length of the driving experience as well as the driving skills. How long this driver is licensed is relevant. Other drivers can be warned that nearby drivers are under training or are newly licensed. Regarding driving skills, we would include categories of vehicles that the driver is licensed to drive and eventual training programs he has completed (hazardous cargo training or any other courses focused on professional drivers).

  Information about elements in the driving experience category can be collected in several ways. User input can be used but is unreliable since drivers can lie or be too optimistic about their skills. Obtaining data by fetching services based on the driver's other identity information can be useful and provide adequate results. Another approach would be using Artificial Intelligence (AI) to infer some of the driver skills based on the driving activities performed, something already viable (JOHNSON; TRIVEDI, 2011).

The previously mentioned contextual element categories would have all the required information to characterize drivers and vehicles. While these are arguably very important components from the point of view of vehicular applications, other components can also play marginal or central roles depending on the requirements and objectives of an application. We will now describe the categories and subcategories defined to contain contextual elements related to cargo (or payload).

Payload identity, in turn, is particularly important when considering commercial vehicles. It specializes in the following categories.

- *Payload Type*: It is defined as a contextual element category and not a single contextual element for the same reasons as the vehicle classification. There are several different methods to classify cargo and applications may need to use more than one classification system at the same time. Particularly, cargo hazard classifications are very useful in a wide range of applications, from inspection to handling emergency situations. Examples are the Globally Harmonized System of Classification and Labelling of Chemicals (GHS) (WINDER; AZZI; WAGNER, 2005) or the United States Environmental Protection Agency Toxicity Category (Environmental Protection Agency, 2021).

- *Payload Dimensions*: these are relevant to applications related to freight.
- *Payload Identification*: this contextual element category holds contextual elements used to uniquely identify the payload. Barcodes, parcel tracking numbers, and any other information that can identify the cargo are part of this category.

Cargo information can be collected by the input of cargo manifest into the system, preferably using integration with other systems to avoid human error. Weight sensors and video or infrared cameras can also be useful to gather information about cargo currently loaded in vehicles.

Device identity is important to handle data from other equipment, such as traffic lights, sensors, and network infrastructure devices. Its first subcategory is *Device Type*, for contextual elements containing information regarding the type and capabilities of the device. Another subcategory is the *Device UID*, for information that can uniquely identify a device, such as a MAC address or a traffic light identification number.

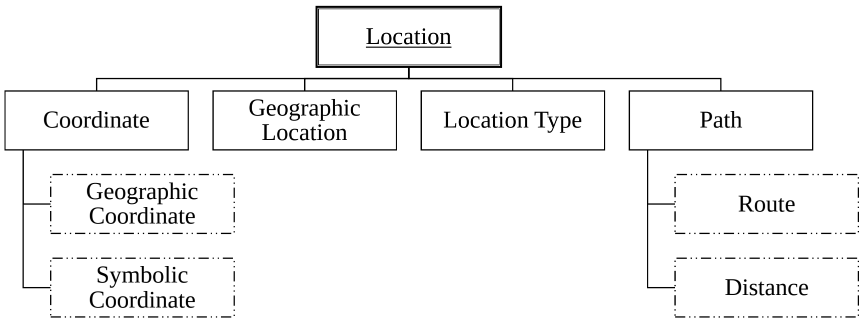
Information used to identify or characterize a place can be categorized as part of the **Place** category. In this sense, the Place identity presents three subcategories as follows.

- *Place Id*: examples of elements in the subcategory are Addresses or institution names (e.g., Museum of Science and Technology).
- *Place Attributes*: this contextual element category encompasses elements such as dimensions of the place, opening hours, and also restrictions such as the maximum allowed weight or height for vehicles to come inside the Place, or the minimum age for people to be admitted to the place.
- *Road Characteristics*: it contains static information about a road section. Its pavement type, number of lanes, length, and other relevant road information fall into this subcategory.

3.4.2 Location

Mobility is a fundamental factor in vehicular applications. These software applications will run most of the time while the cars are in motion, some of them being significant only in this situation. Contextual elements related to vehicle location are the most important for some applications. Location may not seem like a contextual element category with some reasonable diversity of elements, but this is not true. Although geographic coordinates are well-known information to define a location, other ways of defining it also exist and may be relevant in our environment. Figure 6 shows the contextual element categories and subcategories in the Location supra-category.

Figure 6 – Location subcategories of the proposed taxonomy



Source: The author (2022)

Two types of **Coordinates** can be used, each with its own possible contextual elements.

- The first holds contextual elements related to *Geographic Coordinates*. Information like Latitude, Longitude, and Altitude would be classified in this subcategory. The ready availability of sensors that gather data related to the geographic coordinates from GPS and Global Navigation Satellite System (GLONASS) satellites makes their use widespread. Map-based applications are very common and have changed the way drivers get prepared to travel to unknown places. Along with other contextual information described in some other contextual element categories here, applications like Waze, which can help deal with traffic and other commuting issues, have been developed and are heavily used, enhancing the driving experience for millions of people.

- *Symbolic Coordinates* is the other type of coordinates able to define a location in a vehicular environment. This contextual element category is mentioned by Bettini et al. (2010), and Contextual Elements which provide coordinates and identifiers not related to the physical world can be fit here. Examples would be the Cell Id of a cellular network base station, identifiers for other wireless networks, or special purpose beacons placed in strategic locations.

Geographic Location (or Semantic Location) provides more semantics to the location information. Addresses, Road names, floor numbers, and any other information that can be used to identify places without being connected to Geographic features of the location are part of this contextual element category. Generally, the most reliable way to obtain such information is based on geographic or symbolic coordinates and geographic information systems.

Location type: in automotive applications, it can be good to know whether the vehicle is currently on an urban street, on the road, or in a parking lot, for instance. This subcategory serves to identify not the specific and unique place where the vehicle is but the place type. Different rules can apply according to such information. As with the address subcategory, this information often depends on geographic or symbolic coordinate values. Depending on the environment, symbolic coordinates can be more relevant and even be used independently, such as when vehicles are indoors in multi-story car parks.

Path is another location subcategory. It is further divided into *Route* and *Distance*.

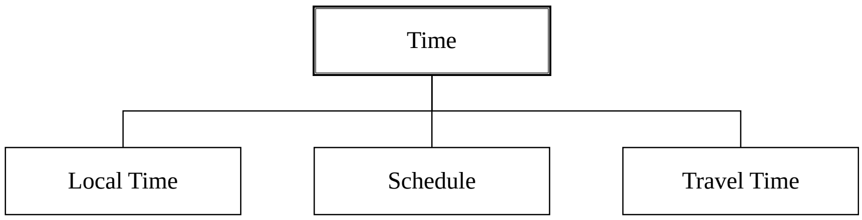
- The contextual elements that are part of *Route* are locations that define a path, from the starting point of the journey until its destination, including both the start and the destination. Manual and automatic alternatives exist to obtain route information. Manual methods include user input to define its route. Automatic methods for obtaining routes are based on user history or connected to web services that contain the user's agenda, bus line information, or passenger current location and desired destination.
- *Distance* holds contextual elements that represent distances between two points, such as the distance between a vehicle and a destination, between vehicles, or between two locations.

These contextual element categories are useful for characterizing the current location of the vehicle as well as its route. In general, location data can be associated with time information to help make the current context of the vehicle clearer.

3.4.3 Time

Time-related contextual elements can be used to refine the identification of the current context. Similar to location, time can be thought of as a simple, indivisible contextual element category. But different types of information can be collected and used based on time values, and we categorize these elements into three contextual element categories: Local Time, Schedule, and Travel Time. It is possible to discuss that the time dimension in context-aware applications is more complex than what is described in this work. However, the taxonomy was built based on the Contextual Elements identified in real projects that were analyzed during this research, and given these Contextual Elements, it was only possible to identify these three subcategories. The organization of the subcategories in this category is illustrated in Figure 7

Figure 7 – Time subcategories of the proposed taxonomy



Source: The author (2022)

Local Time is related to time information of the current vehicle location. This includes the date, time, day of the week, and more subtle or subjective information, such as whether the current day is a holiday, workday, or a weekend day. Applications dealing with traffic information can use it to predict traffic conditions and suggest better alternatives. Timestamps can be collected from local devices' time settings, or more accurately from time servers online. Holiday information can also be consumed from web services.

Schedule is designed to contain contextual elements that represent information of schedule appointments of drivers or passengers, or due dates and times of arrival of the transported cargo. This information can be collected from integration with user's agenda systems (like Google Calendar or smartphone applications), integration with enterprise systems (in case of cargo due dates), historical data, or ultimately but not ideally, user input.

Travel Time is another subcategory, which aims at collecting time information regarding the travel itself. Information like the *time a journey has started* and the *last rest stop* by the driver are in this contextual element category. Such data can be used to measure tiredness probability and recommend drivers to make unplanned stops, for instance. In countries with regulations for maximum continuous driving journey for professional drivers, such information can be used in the inspection of such rules. Together with information from other categories, applications could also suggest rest stops before the driver reaches the legal limits.

3.4.4 Activity

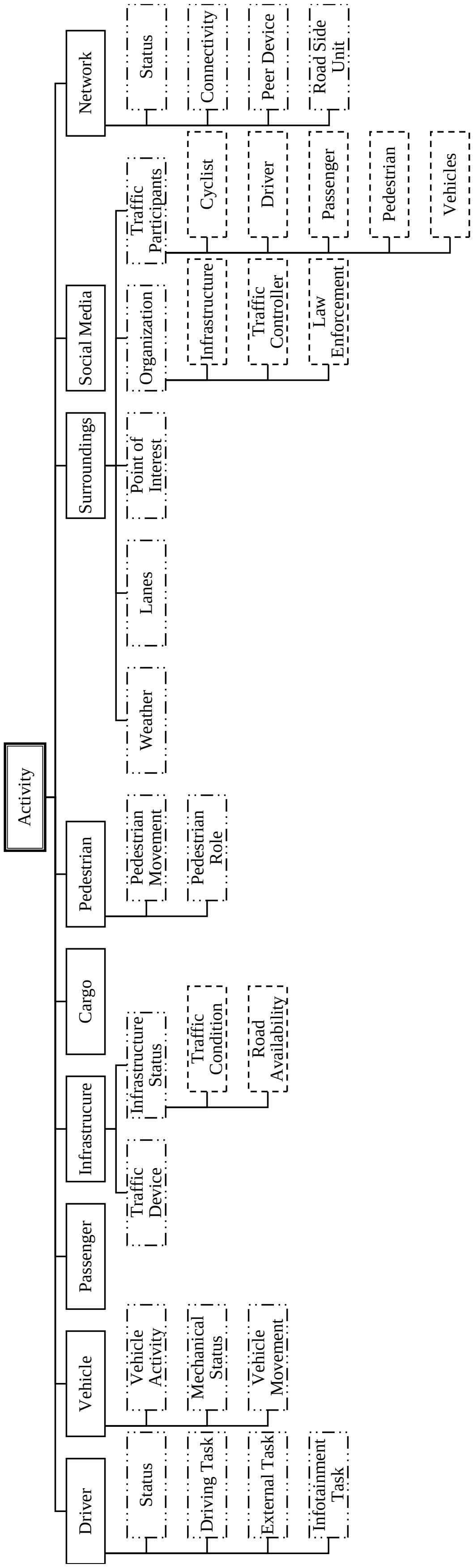
Context-aware applications, in general, have to deal with user and device activities. These activities are performed to execute the tasks, which can be in the current foreground of the application or running in the background, without (or with minimal) user intervention. Vehicular context-aware applications have multiple users and devices contributing to the processing of the current context. Thus, we have defined subcategories for each of the involved components whose activities are useful to such applications. These components include the driver, the vehicle itself, the network connecting the vehicle to other vehicles and devices, passengers, and the surroundings of the vehicle and its route. Figure 8 demonstrates the organization of the Activity category and its subcategories.

The first subcategory we have defined holds contextual elements regarding **Driver**'s activity, as follows:

- *Driver Status* and contextual elements in this category can be used, for instance, to measure driver's attention, tiredness, and other important information related to the tasks a driver is executing while driving. Biometrics data such as pulse rate, body temperature, or blood alcohol content can also be part of this contextual element category. When combined with contextual elements from other categories, driver activity data enables applications to deliver information on adequate media, affecting driver's concentration as low as possible.

- *Driving tasks* are expected to be the most common type of tasks handled by drivers. Driver's actions on pedals, wheel, gears and every other car interface used to control the vehicle are low-level data that can be considered as contextual elements for this category. Such raw data can be retrieved through a vehicle bus like On-Board Diagnostics II (OBD-II) or specific sensors in each of the interfaces. More advanced low-level data can be collected using cameras, which could perform eye-tracking to identify the direction where the driver is looking at. Using inference techniques and combining with other data,

Figure 8 – Activity subcategories of the proposed taxonomy



Source: The author (2022)

applications could generate high-level contextual elements which fit in this category, such as the identification that a driver is performing an evasive maneuver, emergency braking, or parking, for example. Driver concentration level can be considered as another high-level contextual element that can be computed using other data as input and fits in this category.

- *Infotainment tasks* are related to the driver actions related to devices providing information and entertainment. While not their main task, drivers commonly have to deal with equipment like vehicle radio, multimedia center, or GPS devices. Nonetheless, there is also high-level information in this contextual element category, such as knowing which station is tuned or which route on GPS device is being followed.
- *External tasks* is a contextual element category to hold any activity a driver is performing that is not related to driving or handling any other device in the car that controls information or entertainment systems. Their interaction with passengers or equipment that are not part of the vehicle is in this contextual element category.

Passenger's activity is another subcategory of the Activity category. Contextual elements representing the current activity and status of passengers would fit in here. It would have contextual elements to define the number of passengers on board, their current seats, and other information that identify not a particular passenger, but the group of passengers currently in the vehicle. Their actions, objectives, and tasks are useful information to infer current context. Also, dynamic information about the passenger, such as biometric data like body temperature and pulse rate, are useful to some applications. Gathering values for these elements is probably the most difficult of all the contextual element categories in this model, since passengers usually have very low interaction with vehicle controls or interfaces. Using cameras, image and motion recognition, presence or weight sensors, and combining other information is required to obtain valid and useful values for the elements in this subcategory.

The **Pedestrian** contextual element category is divided into two subcategories: *Pedestrian Movement*, which holds elements such as the speed, direction, and acceleration of pedestrians nearby the vehicle, and *Pedestrian Role*, that holds elements that define the role of that pedestrian in the ITS, such as whether they are potential future passengers or if their activity is relevant to the system, such as a traffic agent or a first responder.

After describing the person-related activity contextual element categories, we define the

Vehicle activity contextual element category. The category is subdivided into vehicle *activity*, *movement*, and *mechanical status*.

Information concerning the tasks the vehicle itself is performing, as well as the metrics for some variable information regarding the vehicle are contextual elements related to the *Vehicle Activity* subcategory. Some driver activities can trigger changes in values of elements in the Vehicle Activity category, such as a Taxi driver can accept a ride in the taximeter (thus, he is performing an interaction with a device of the vehicle that is not essential to driving, which itself is part of the Driving External Task), and this action will result in a change in the "Is on duty" property of the vehicle (part of the Vehicle Activity category), since now the taxi is no longer free. Another important part of *Vehicle activity* is contextual elements that provide information regarding what the vehicle is being used for. While these elements might seem to be the same as those in the Identity/Vehicle/Classification contextual element category, their contextual elements are different. Vehicle classification is static, while the type of service the vehicle is providing is dynamic. A pickup truck does not change its classification as a light-duty vehicle, but the same vehicle might be used, at different moments, for transporting passengers, for emergency situation handling, or for cargo hauling, for instance. Another example is the service status of a bus, whether it is in or out of service. Collecting data that can be used to infer the current service can be challenging. The inference from other contextual elements can be a valid approach, as values for some possibly present sensors could be used to identify the service.

Vehicle Movement subcategory holds information that can describe the motion attributes of a vehicle. Contextual elements which might fit in this category are speed, direction, and acceleration. Speed can be easily obtained using the vehicle's OBD-II interface. Acceleration (not to be confused with the pressure on the throttle pedal) can be calculated using distance, speed, and timers. Direction can be obtained from modern GPS receivers.

Mechanical Status is the subcategory to hold most of the information that can be collected from vehicle data buses as OBD-II or sensors on car parts. Interfacing with default buses to obtain data is not hard, but collecting information from sensors in parts not connected to such infrastructure can be challenging. Status messages that show that maintenance is required or the extent to which the vehicle has been handled can be higher-level information about mechanical status.

The **Cargo** Activity contextual element category holds contextual elements that characterize the payload interaction with the vehicle, either when already loaded or while waiting to be loaded. The temperature of the cargo, in the case of perishable materials, and pressure for transport of gases are some of the elements in this contextual element category. Monitoring data of livestock being transported is another example.

Social Media Activity is a contextual element category to hold contextual elements regarding data coming from social media. The usefulness of data obtained from social networks, such as Twitter, to predict traffic jams and other transit-related issues is well accepted, and various researches have already been performed in such a direction (WONGCHAROEN; SENIVONGSE, 2016; ESSIEN et al., 2020). Information gathered from social media regarding friends and acquaintances nearby is also useful in some ITS applications.

The **Network** Activity contextual element category contains contextual elements which represent the state of the network that a vehicle is using to communicate with other vehicles - Vehicle-to-Vehicle (V2V) -, road infrastructure - Vehicle-to-Infrastructure (V2I) -, or the Internet, and as a consequence, with any other connected device - Vehicle-to-Everything (V2X). In this contextual element category, the *Connectivity* subcategory holds information about the network, such as bandwidth, type of network, and level of connectivity. The *Status* subcategory contains information about network statistics that are not part of the Connectivity subcategory. *Road Side Unit* holds contextual elements representing information collected from RSUs in the same network of the vehicle, such as traffic flow on a road segment (WOODARD; WISELY; SARVESTANI, 2016). *Peer information* would hold contextual elements about the peers, the kind of device they are, and their interfaces to obtain more information.

The **Infrastructure** contextual element category is meant to hold information regarding the status of road-related equipment and the road itself. It is further divided into two subcategories, *Traffic Device* and *Infrastructure Status*.

- The *Traffic Device* subcategory holds information about road equipment such as traffic lights, messaging boards, traffic signs, toll plazas, and other road devices. Such equipment have data that can be very helpful to many vehicular applications. Traffic lights can share their current color and how long would take for it to change, message boards could broadcast their current message or more detailed information that would otherwise be not feasible to be displayed due to its size restrictions. Moreover, traffic signs could share their enforcing rules or warnings to vehicles without needing to rely on online databases that can be not updated, making sure that vehicular applications receive the same data as the driver can see. V2I communication is a viable solution to obtain road device information.

- *Infrastructure Status* contains two subcategories. ***Traffic Condition*** is a contextual element category with contextual elements to ITS applications, such as the level of traffic jam and its causes (such as cars stopped on the road, potholes and other events that can impact the traffic flow). ITS applications can use this kind of information regarding the traffic and road conditions to infer current context and predict future situations. The other subcategory of Infrastructure Status is ***Road Availability***, which contains elements regarding the possible blocks (either total or partial) on the road. Indeed, applications such as Waze make heavy use of information from both *Traffic Condition* and *Road Availability* to provide driver assistance. Some data related to road availability and traffic conditions can be collected from devices broadcasting the road status, inferred through cameras and other sensors, or obtained through web services. Such services can be kept updated by using crowdsourcing techniques, as used in the already mentioned Waze application.

The last subcategory in the Activity category is related to the **Surroundings** of the vehicle. This subcategory is a parent to several other subcategories that will be further described. Information can be collected by using wireless networks or the integration of GPS data with online services. Cameras, Radio Detection and Ranging (RADAR), and Light Detection and Ranging (LIDAR) sensors can also be used to gather data about the surroundings of the vehicle. This contextual element category not only knows which vehicles, people, or points of interest are in the surroundings but mainly knows of their current activities. For example, understanding that a pedestrian close to the vehicle is on a cell phone, potentially distracted, is crucial to avoiding an accident; having information on the type and activity of nearby vehicles can be very useful if there are emergency vehicles in the vicinity; knowing that a nearby restaurant is currently open is an interesting information for the users of a vehicular application.

Weather is the first subcategory in the Surroundings subcategory. Information regarding temperature, wind, air humidity, and rain or snow forecasts is very useful to many ITS applications. Also important in the Surroundings domain is the *Lanes* subcategory. It holds information about the number, availability, and current way of lanes (in case of reversible lanes) in the vicinity of the vehicle. Another Surroundings subcategory is *Point of Interest* (PoI). It holds information about any location in the vicinity of the vehicle or its route that might be useful to the context of the application. Gas station fuel prices, tourist attraction information, or stores in the route are possible information of this contextual element category that are used in ITS projects.

Two of the Surroundings subcategories are further subcategorized as shown in Figure 8: *Organization* and *Traffic Participants*. *Organization* has the ***Infrastructure*** subcategory, which holds information about road infrastructure near the vehicle, i.e., this subcategory focuses on the presence and status of infrastructure items only in the vicinity of the vehicle, in contrast to the *Activity/Infrastructure/Status* subcategory. ***Traffic Controller*** holds information about entities that have the power to control the traffic flow, such as traffic agents or traffic signs near the vehicle or on its route. Location is a key differentiating factor when such information is in this contextual element category or in *Activity/Infrastructure/Traffic Device*. The state of a traffic sign is always part of the *Activity/Infrastructure/Traffic Device*, but if that traffic sign can directly affect the vehicle, it is also part of the *Traffic Controller* subcategory. ***Law Enforcement*** regards the presence and role of police or traffic agents, speed cameras, and other entities involved in traffic law enforcement. While a common (albeit controversial) use of elements in this contextual element category is to warn drivers of the presence of these entities on their route, less disputed uses of elements in this category exist, such as automated first-responder allocation systems and other security and safety applications.

The *Traffic Participants* subcategory includes ***Cyclist***, ***Driver***, ***Passenger***, ***Pedestrian*** and ***Vehicle***. Their elements are both the presence of any of these participants in the vicinity as well as any other activity information relevant to the system regarding one of those participants.

3.4.5 History

Historical data can have a multitude of uses, being helpful in predicting future situations based on previous ones. Knowing the traffic intensity information for a long period can help to predict future traffic. Previously captured data about fuel consumption, location and several other information are useful and are already used in applications. Every previously mentioned contextual element from the aforementioned categories could be stored if helpful in some context to an application. Such accumulated data can also be used to infer why some previous activities and outcomes happened, getting reasoning on which the application can base itself to better identify and adapt to future context.

Major issues related to contextual elements in this category are not related to gathering data, but with their storage. Depending on the granularity, a large amount of data can be generated, making local storage alternatives unfeasible, bringing us to cloud storage as a viable alternative.

Still, we have to be concerned with privacy. Historical information is dangerous in the wrong hands, so data security is essential if a system would require storing such data online. When correctly used, it can have multiple good outcomes besides the ones already mentioned.

Historical use of contextual elements was not specified in terms of categories in this work, since we have observed that all history-related information is also part of some subcategory of the other supra-categories.

Those are the categories defined in our model. While we strongly believe that these categories reflect most of the existing useful contextual elements to the vehicular application domain, we reinforce that this model is not exhaustive, so it can be extended.

3.5 EVOLUTION AND VALIDATION

Three connected approaches were used to validate and evolve the taxonomy from its initial forms until the current proposed taxonomy. The first approach was a blind experiment where software development professionals were assigned the task of designing a context-aware vehicular application in a particular scenario. The second approach was to proceed with the complete process of designing and developing a vehicular application using this taxonomy in the process. Finally, with a mature iteration of the taxonomy, we used it to build a knowledge base of existing ITS projects in the literature.

3.5.1 Blind experiment

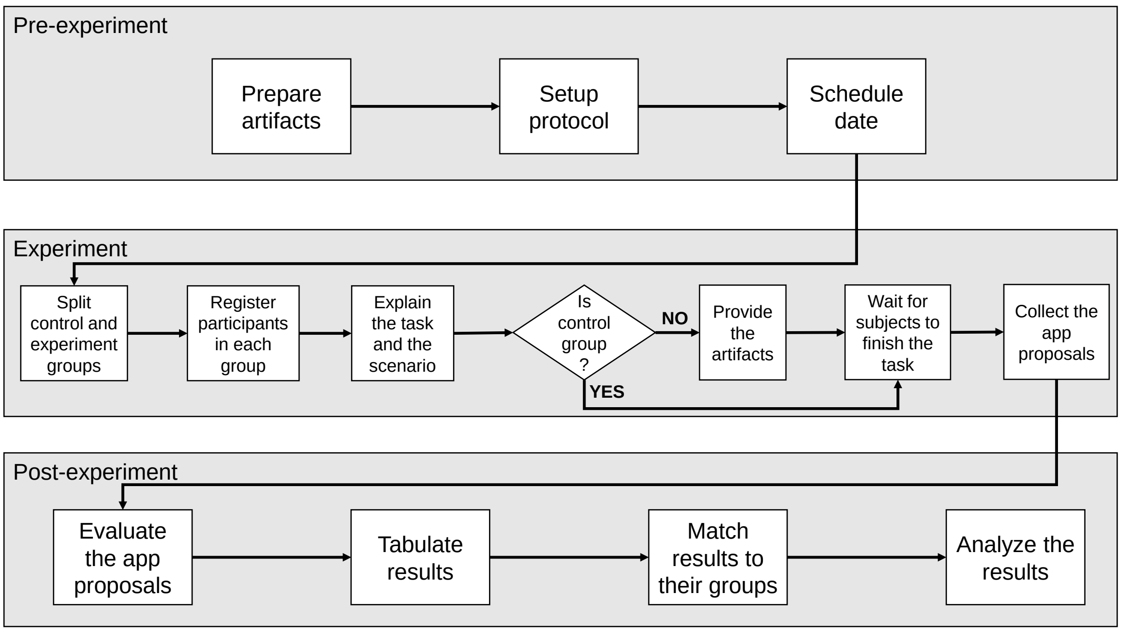
In this experiment, we had the participation of 21 subjects, in two rounds. In each round, the concepts of ITS, vehicular applications, context-awareness, and contextual elements were introduced to all subjects. An explanation of the scenario was given. The chosen scenario was a vehicular application to improve the effectiveness of service provided by emergency vehicles, such as ambulances, fire engines, or police cars.

The first round was performed with twelve graduate students with moderate professional software development experience. The students were enrolled in the Distributed Systems course. Some of them had previous experience with Computational Context, however ITS was not a common knowledge to them.

The second round was performed with nine researchers, members of the Laboratório de Inovação Veicular (LIVE)[1] research project, focused on investigating innovative solutions for vehicular applications, connected to one of the largest car manufacturing companies in the world. As such, the knowledge of this group of subjects in ITS was high. Their formal knowledge in context-awareness related topics, on average, was existent but not quite as good as that of the subjects who participated in the first round.

Figure 9 shows the process used in this evaluation. The same process was used for both rounds. After the initial preparation of the experiment, with the definition of the protocol to be followed and the creation of the artifacts that would be delivered to the subjects in the experiment group, we proceeded to execute the experiment. All subjects were briefed on what they were expected to accomplish: the overall designing of a coherent application in the described scenario, with the description of features that could be desirable, and mentioning which information the application would require to properly work. Participants were advised not to be constrained by what they think is currently possible with existing or deployed technology, with the sole restriction being the application being viable and coherent.

Figure 9 – Process of the blind evaluation of the taxonomy



Source: The author (2022)

Then, in both rounds, they were divided into two groups, each in a separate room. The first was a control group, where participants had no contact with the proposed taxonomy (initial

[1] Laboratory for Vehicular Innovation

version). They were allowed to start immediately designing the application. The second group was presented with the taxonomy and a table abstractly describing each category, as presented in this chapter. A very brief explanation was given to this second group about the hierarchical organization of the taxonomy, and then they were allowed to start designing the application.

Each participant was given one and a half hours to individually design the application according to informed rules. Subjects were free to format the output in the way they preferred, but it was recommended that a brief description of the purposes of their application be given first, and then a list of features. For each feature, the participant gave short descriptions of why it would be useful, who would benefit from such a feature, input data that would be required, and output data that the feature would generate.

Subjects were instructed not to identify the group they took part in the responses form, so an unbiased evaluation could be possible, with each response being identified only by a code. The list of participants in each group was hidden from the researchers until the end of the evaluation.

Two different researchers evaluated the proposals. Each researcher needed to assess a 0-10 score to five aspects of the application: *Coherence*, *Usefulness*, *Number of distinct contextual elements used*, *Number of distinct contextual elements incorrectly used*, and *Viability under current technology*. The *Coherence* was evaluated based on how the proposed application was in adherence to what the proposed scenario requested, and was given a score from 0 to 10. The *Usefulness* was evaluated according to the value of the proposition to the proposed scenario, such as how much positive impact it would have if it were developed and deployed, and was also given a score from 0 to 10. *Viability* was measured according to, given the current technology, how feasible would be the development and deployment of the proposed application. It was also scored from 0 to 10, where 0 means completely unfeasible with current technology, and 10 is totally feasible with current technology. The *Number of distinct contextual elements used* is straightforward, and the *Number of distinct contextual elements incorrectly used* is the number of contextual elements that were indicated by the subject in the design of his/her application, but that was neither obvious on how it could be used or was not explained how the subject would use that contextual element to help on characterizing context in that application.

The results of the experiment are available in Table 1. The overall application *usefulness* was evaluated as 19.5% higher in the group that used the taxonomy when compared to the applications idealized by those in the control group. Due to the small sample, however, it is not possible to affirm that the difference is statistically significant (Mann-Whitney $U = 75$,

$n_1 = 11$, $n_2 = 10$, $p = 0.1603$ two-tailed). This can be checked by the high p-value for this dimension in Table 1. To have a 95% confidence that the results are different, the p-value must be ≤ 0.05. For the same reason, no significant difference was found in the *viability under current technology* (Mann-Whitney $U = 46.5$, $n_1 = 11$, $n_2 = 10$, $p = 0.5634$ two-tailed) and *coherence* (Mann-Whitney $U = 65.5$, $n_1 = 11$, $n_2 = 10$, $p = 0.4662$ two-tailed) of the application aspects.

Regarding the number of contextual elements used, the results were much higher in the applications of the experimental groups. We used the *number of distinct contextual elements used* and the *number of distinct contextual elements incorrectly used* to calculate another metric, the *correctly used contextual elements*, as a simple subtraction of the number of incorrectly used contextual elements from the total number of contextual elements used. While applications of the control groups averaged 4.5 *correctly used contextual elements*, applications designed by subjects in the experimental groups averaged a much higher 12.2 correctly used contextual elements per application. This indicates that the taxonomy of contextual element categories helped to raise engineers' awareness of the possibility of using different contextual elements. In both of these dimensions, it is possible to affirm with more than 99% of confidence that the difference in the average number of contextual elements obtained from the control and experiment groups is statistically significant (Mann-Whitney for the number of distinct contextual elements used is $U = 95$, $n_1 = 11$, $n_2 = 10$, $p = 0.005291$ two-tailed, and for the number of distinct correctly used contextual elements the values are $U = 94.5$, $n_1 = 11$, $n_2 = 10$, $p = 0.005816$).

The choice for the Mann-Whitney U Test comes from our small sample size (each group with $5 \leq n \leq 20$), since this tests performs correctly on such small samples. Since our sample values are not guaranteed to follow a normal distribution, the Mann-Whitney U Test is also a good fit, since it does work with any distribution. While it essentially compares medians instead of means (as would be the case for a t-test), it still is able to answer our question of whether the results of the groups are significantly different. Two-tailed tests were performed, since we need to know whether there is any difference, not only if one group is higher than the other.

This validation also served to improve the taxonomy, since categories referring to traffic devices, conditions, and surrounding traffic participants were suggested by the subjects of the experimental groups and. After validation with the projects in the knowledge base, these suggestions were added to the taxonomy.

Table 1 – Overall results of the experiment (both rounds aggregated)

	Usefulness	Viability	Coherence	# Distinct CEs	# Correctly used Distinct CEs
Control	7.0	7.6	8.2	4.5	4.1
Experiment	8.4	7.2	8.5	12.5	12.2
Difference	+19.5%	-5.5%	+4.2%	+178.8%	+197.1%
P-value	0.1603	0.5634	0.4662	0.0053	0.0058

Source: The author (2022)

3.5.2 Design and development of a vehicular application using the proposed taxonomy

We designed an ITS application using the taxonomy as input to aid in defining the application's context-related features. CONVOY (Context-Oriented Navigation of Vehicles On the waY) is a system intended to assist drivers taking part in car groups to follow the same path, navigating through roads and traffic, and reacting accordingly to unforeseen events which might happen.

One of the drivers is designated the leader who must define the route that the other vehicles in the group must follow. Two options to define the leader can be made: Driver (using elements of the category *Identity/Person/Identification/Driver Id*) or Vehicle (using elements of the category *Identity/Vehicle/Vehicle Id*). In addition, different versions of the CONVOY application can suggest the most suitable driver to be the leader on a journey, using information from contextual elements from the categories *Identity/Person/Information/Driver Experience* and *Identity/Person/Information/Profile*.

The leader must also choose when the convoy should stop to rest, eat, or sightseeing (*Activity/Surroundings/Points of Interest*). CONVOY should assist group leaders by notifying optimal times to rest or stop for eating, assessing values of contextual elements from several categories. Values from *Location/Path/Route*, *Location/Coordinate/Geographic Coordinates* and *Time/Schedule* can be used to define the distance to the destination, relate it to the passengers' schedules, and infer whether it would be worth stopping. Information from the *Time/Travel Time* and *Identity/Person/Information/Physical Attributes* categories can help prevent tired drivers from being kept on the road. The use of *Identity/Person/Identification/Passenger Id* and *Identity/Information/Profile* can also help if children, pregnant women, people with disabilities or any type of passenger with special needs are present in the group, so that stops can be scheduled accordingly with their needs. Not least, the use of *Activity/-Surroundings/Points of Interest* and historical data can help identify safe places to rest. An ideal application could merge information from all the aforementioned contextual elements to decide the best time to suggest a stop for resting or sightseeing.

CONVOY also focuses on sharing the following information among the vehicles in a group:

- Position of each vehicle, with an adaptive map that zooms in to fit the vehicles in the viewing window according to contextual information (*Location/Coordinate/Geographic Coordinates*, *Activity/Vehicle/Movement*, and *Location/Path/Route*);

- Vehicle status, such as whether it is moving, stuck in slow traffic or in emergency (using elements in the *Activity/Vehicle/Movement*, *Activity/Vehicle/Mechanical Status*, *Activity/Driver/Status* and *Activity/Passenger* categories);

- Route change notifications (*Location/Coordinate/Geographic Coordinates* and *Location/Path/Route*);

- Warning when a vehicle is too slow compared to the leader (*Activity/Vehicle/Movement*).

- Notification that the leader is too fast when compared to other vehicles in the group (*Activity/Vehicle/Movement*).

Every participant can declare an emergency, which would be due to contextual elements in the *Activity/Vehicle/Mechanical Status*, *Activity/Driver/Status* or *Activity/Passenger* categories.

CONVOY can be used as an application to be embedded in automotive systems. Concerning non-functional requirements such as performance, being aware of contextual elements in the *Activity/Network/Connectivity* category can improve application performance and resource usage.

The same concepts about the envisioned application were also given to another similarly experienced engineer, but not providing him our model. The result of his design process presented a much smaller context-enabled feature set. This developer elicited the use of maps in a similar way thought by the other professional who had the support of our taxonomy, but none of the other features appeared in his design. All the features he envisioned regarding the map usage were also covered by the design project given by the analyst who used the proposed taxonomy.

In summary, CONVOY used 17 contextual element categories distributed as follows: 7 related to the Identity supra-category, 2 related to the Location supra-category, 2 related to the Time supra-category and 6 related to Activity.

3.5.3 Creation of a knowledge base of contextual element categories used in projects available in the literature

The final taxonomy validation step was the creation of a Knowledge Base (KB) from ITS projects available in the literature. This KB should list the contextual element categories of the taxonomy, which are used in the projects in question. Therefore, we would validate the practicality of the categories defined in the taxonomy since there must be at least one project in the current literature that uses elements from each of the defined categories.

The 70 projects identified in the literature review performed during the design of the taxonomy use contextual elements from at least one of the contextual element categories defined in our taxonomy. Furthermore, three worldwide used applications, Waze, Uber, and Moovit, were also analyzed and included in the knowledge base, leading to 73 projects in total. Waze and Uber were analyzed based on the author's personal experience as user of both applications, supported by full navigation through their features. In the case of Moovit, alongside the navigation of the application, an article that contains a section describing Moovit helped on mapping the categories used in the application (SANTOS; NIKOLAEV, 2021), to compensate for the author's lack of experience on using the application. All categories in our proposed taxonomy appear in at least one project in this knowledge base.

It is important to observe that while the process of identifying ITS projects has been presented here as a single consolidated step, it was not done all at once. During this research, the taxonomy had several iterations where the list of ITS projects that were analyzed has been reviewed and updated. This process and the projects identified through it were both used to define the knowledge base described in this section, and to define the categories themselves that are part of the taxonomy, as mentioned in the 3.4 section.

During the development of the knowledge base, we did not map whether the usage of elements from a category is for current or historical data. While this information could be valuable for some potential uses of this knowledge base, our objective is to validate whether the categories in the taxonomy are indeed used in real projects. Thus, we marked the category as used in a project either when its data usage was current, historical, or both.

The format of the knowledge base is a simple binary matrix that represents the relations between projects and the contextual element categories used in them. The value 1 in a cell represents that the project of the cell's line in the matrix uses at least one contextual element from the category of the cell's column. A sample of this matrix is shown in Frame 5, while the final version of the knowledge base is available in Appendix D.

Frame 5 – Sample of the knowledge base binary matrix

	Local Time	Schedule	Travel Time	...
Silva, Borges e Vieira (2018)	0	0	0	...
Vieira, Caldas e Salgado (2011)	1	1	0	...
Kannan, Thangavelu e Kalivaradhan (2010)	0	0	0	...
Johnson e Trivedi (2011)	0	0	0	...
Meier, Harrington e Cahill (2006)	1	1	0	...
Aguirre et al. (2017)	0	0	0	...
...	...	...	...	...
Moovit (SANTOS; NIKOLAEV, 2021)	1	1	1	...

Source: The author (2022)

This knowledge base can be used for several objectives. On the validation of the taxonomy proposed in this research, it was useful to check that every category proposed was used in at least one project, making sure that they are valid categories for ITS applications.

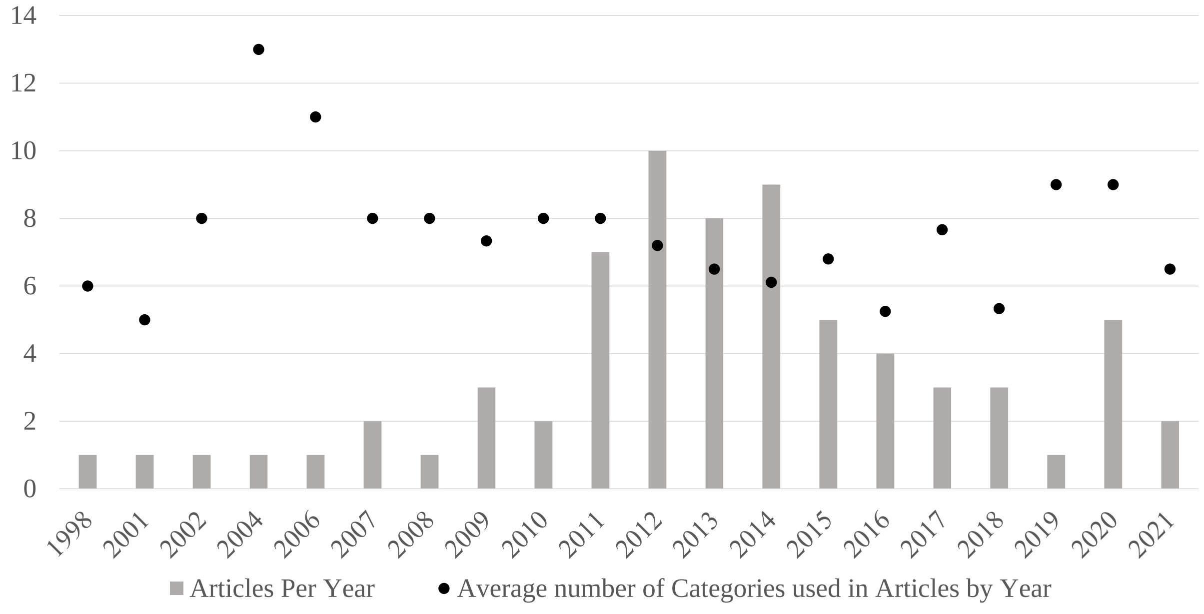
Another possible use of the knowledge base is for system designers to check for applications similar to the one that they are working on. It is possible that, given an incomplete subset of known contextual elements categories that will be used in the new application, the designer will look for projects that use a similar subset of categories and check whether they use any other category.

From the aforementioned possible usage of the knowledge base, we observed the potential to create an automated tool to perform not only the direct observation of similar projects, but which could also improve upon that by attempting to identify the most probable categories which could also be useful to the newly designed application. This work resulted in the recommender system described in Chapter 4

Furthermore, it is possible to perform some analysis with data available in the knowledge base. Figure 10 shows the distribution of projects per year and the average number of categories used by the projects each year. It does not take Waze, Uber and Moovit into account as it is not possible to map when the current feature set of each of these projects has been defined. We highlight the period between 2011 and 2017, which concentrates most projects in the knowledge base. During this period, the average number of categories in a project ranges from 5.25 to 8. The average number of categories used per project for the whole knowledge base is 7.2.

In addition, Table 2 shows the top-10 categories used by the projects in the KB. Geographic coordinates representing the location of an object ranks number 1, being used by 53.4% of the projects in the KB. This is expected, due to the high dependency that, in general, ITS applications have on location data, especially the exact position where the entities are in the world. Indeed, there is no surprise in the data of this table, since all the highest-ranked categories are obviously demanded in ITS applications.

Figure 10 – Distribution of projects per year and the average number of categories used in the projects of the articles of each year



Source: The author (2022)

The aggregation of how many contextual element categories each of the analyzed projects use, grouped by the supra-categories *activity*, *identity*, *location*, and *time*, can be observed in Table 3. An observation that can be made of the data in this table is the diversity of the projects regarding their use of contextual element categories according to the proposed taxonomy. While some projects, such as Chen e Lu (2015) only use contextual elements from a single category, other projects, such as Hu et al. (2015) use elements from 15 different

Table 2 – Top-10 categories used in the 73 analysed projects.

Rank	Category	% of projects that use the category
1	Location/Coordinate/Geographic	53.4%
2	Activity/Vehicle/Movement	47.9%
3	Time/Time	41.1%
4	Activity/Infrastructure/Status/Traffic Condition	32.9%
5	Activity/Surroundings/Traffic Participants/Vehicle	31.5%
5	Identity/Vehicle/Vehicle Id	31.5%
7	Identity/Person/Information/Profile	30.1%
8	Activity/Surroundings/Weather	28.8%
8	Identity/Vehicle/Classification	28.8%
10	Time/Schedule	26%

Source: The author (2022)

categories. Another noteworthy observation is the fact that only two projects do not use any contextual element category from the Activity supra-category (Atasoy et al. (2015) and David et al. (2013) - Bus Stop). This fact is not replicated in the other supra-categories, where each of them have several projects that do not use any element from their subcategories.

Still analyzing data in Table 3, it noticeable that Waze and Uber use contextual elements from far more categories (30 and 25, respectively) than the average of the projects found in the reviewed articles, i.e., about four times more than in the other projects. Two hypotheses have been raised: 1. Mature commercial products evolve, get richer in features, and naturally use more contextual elements, hence, more categories; 2. Research projects might use more contextual elements than it has been possible to capture from their publications. Further research could check whether any of these hypotheses are valid.

On a final note, it is important to reinforce that the usage of the term *knowledge base* to describe this data is correct according to the definition of Waterman (1986) that, when discussing Expert Systems, the "collection of domain knowledge is called the *knowledge base*". However, the artifact that we are referring to as the knowledge base also fits the definition of a *dataset*. According to Nasution, Nasution e M. (2020), "a dataset is a collection of data objects, namely records, points, vectors, events, cases, samples observations or entities". This definition perfectly matches our knowledge base. Also, the data is made publicly available in this work and can be reused for other research projects which might make use of data on the

Table 3 – Number of contextual element categories used in each of the 73 analysed projects. (Numbers in parentheses are the total number of categories per supra-categories.)

(continua)

#	Reference/Project	Activity (30)	Identity (18)	Location (6)	Time (3)
1	Silva, Borges e Vieira (2018)	2	0	1	0
2	Vieira, Caldas e Salgado (2011)	2	2	4	2
3	Kannan, Thangavelu e Kalivaradhan (2010)	10	3	0	0
4	Johnson e Trivedi (2011)	1	0	0	0
5	Meier, Harrington e Cahill (2006)	2	3	4	2
6	Aguirre et al. (2017)	8	2	0	0
7	Younes, Boukerche e Mammeri (2016)	3	1	0	0
8	Sukode e Gite (2015)	3	0	0	0
9	Chen e Lu (2015)	1	0	0	0
10	Sujitha e Punitha (2014)	2	3	0	1
11	Zardosht, Beauchemin e Bauer (2014)	4	2	2	1
12	Khekare e Sakhare (2012)	2	0	1	0
13	Maslekar et al. (2011)	3	0	3	0
14	Fogue et al. (2011)	3	6	1	1
15	Ghaffarian, Fathy e Soryani (2012)	3	1	1	0
16	Elhadef (2015)	3	1	2	0
17	Alazawi et al. (2011)	6	1	2	1
18	Bergan, Bushman e Taylor (1998)	2	4	0	0
19	Alhammad, Siewe e Al-Bayatti (2012)	2	7	2	1
20	Bae e Olariu (2010)	3	0	0	0
21	Alghamdi, Shakshuki e Sheltami (2012)	5	0	1	1
22	Ramesh, Vidya e Pradeep (2013)	5	1	1	0

Table 3 – Number of contextual element categories used in each of the 73 analysed projects. (Numbers in parentheses are the total number of categories per supra-categories.)

(continuação)

#	Reference/Project	Activity (30)	Identity (18)	Location (6)	Time (3)
23	Al-Sultan, Al-Bayatti e Zedan (2013)	7	0	2	1
24	Zarza et al. (2013)	3	0	1	0
25	Wang, Jiang e Mu (2013)	4	3	1	1
26	Fuchs, Rass e Kyamakya (2008)	6	2	0	0
27	Woernd e Eigner (2007)	3	4	3	0
28	Alghamdi (2012)	3	0	1	0
29	Hoogendoorn, Breukink e Arem (2012)	2	0	2	0
30	Ngai et al. (2012)	2	6	2	2
31	Baltrunas et al. (2011)	4	3	1	1
32	Raphiphan et al. (2009)	2	0	1	2
33	Rico et al. (2013)	3	1	2	2
34	Ramazani e Vahdat-Nejad (2014)	2	0	1	1
35	Nassar, Kamel e Karray (2016)	2	3	0	0
36	Rauscher et al. (2009)	4	4	1	0
37	Zhang, Cheng e Lin (2012)	1	0	0	0
38	Bifulco, Amitrano e Tregua (2014) - Singapore	2	2	1	0
39	Bifulco, Amitrano e Tregua (2014) - Amsterdam	1	2	0	0
40	Barba et al. (2013)	4	0	1	1
41	Arkian, Atani e Kamali (2014)	6	1	1	0
42	Santa e Gómez-Skarmeta (2009)	6	1	1	0
43	Panagiotopoulos e Dimitrakopoulos (2019)	5	3	0	1
44	Figueiredo et al. (2001)	3	0	0	2

Table 3 – Number of contextual element categories used in each of the 73 analysed projects. (Numbers in parentheses are the total number of categories per supra-categories.)

(continuação)

#	Reference/Project	Activity (30)	Identity (18)	Location (6)	Time (3)
45	Atasoy et al. (2015)	0	4	2	3
46	Hu et al. (2017)	2	3	1	1
47	Kolbe et al. (2017)	2	3	1	0
48	Parodi et al. (2016)	2	0	3	1
49	Subramanyam e Kumar (2016)	1	2	2	1
50	Hu et al. (2015)	10	2	2	1
51	Nakamura et al. (2014)	1	2	3	1
52	Narayanan et al. (2014)	2	2	1	0
53	Wang, David e Chalon (2014)	5	3	0	0
54	David et al. (2013) - Loading zone	2	0	0	2
55	David et al. (2013) - Bus stop	0	3	1	0
56	Parundekar e Oguchi (2012)	2	4	3	1
57	Werther e Hoch (2012)	5	4	2	3
58	Saha e Chaki (2011)	3	4	2	0
59	Sadoun e Al-Bayari (2007)	1	4	1	0
60	Gena e Torre (2004)	6	4	2	1
61	Goto e Kambayashi (2002)	1	3	2	2
62	Mondal e Rehena (2021)	3	0	2	1
63	Chavhan et al. (2021)	6	0	1	0
64	Jiang et al. (2020)	10	0	0	1
65	Tao et al. (2020)	1	1	0	0
66	Chavhan et al. (2020)	5	10	2	3

Table 3 – Number of contextual element categories used in each of the 73 analysed projects. (Numbers in parentheses are the total number of categories per supra-categories.)

(conclusão)

#	Reference/Project	Activity (30)	Identity (18)	Location (6)	Time (3)
67	Özkul, Capuni e Domnori (2018)	3	2	2	0
68	Ali, Muhammad e Khan (2020)	3	0	1	0
69	Haque et al. (2018)	1	0	2	3
70	Sabet et al. (2020)	2	2	2	2
71	Waze	17	5	5	3
72	Uber	10	8	4	3
73	Moovit (SANTOS; NIKOLAEV, 2021)	3	3	4	3

Source: The author (2022)

usage of contextual elements on ITS projects, and such reuse is also a common characteristic of datasets. While we will keep using the *knowledge base* term to describe it, consider that in this research, it could be used interchangeably with the term *dataset*.

3.5.4 Versions of the taxonomy and of the knowledge base

The knowledge base that is shown in section 3.5.3, and consequently the taxonomy that was used to build it, are the result of the last iteration of the evolution process described in the Figure 2. Several iterations occurred during the design, evolution, and refinement of both the taxonomy and the knowledge base, and three particular of these iterations are important to be described.

- **Version 0.1:** This was the first draft of the taxonomy, that used 15 articles of ITS projects or surveys related to the ITS domain to define a taxonomy with 50 categories (CHAGAS; FERRAZ, 2017). While this number might seem near the final result found in the last version, it is relevant to notice that during the evaluation of this version of the taxonomy, some of the categories that were defined were removed, because they were not categories in fact, but simply a single Contextual Element.

- **Version 1.0:** This was the first stable version of the taxonomy. We consider that it is stable because it has a set of categories that changed little in the iterations that happened before it, and its knowledge base consists of a good number of 61 projects. These projects are quite diverse and make a good representation of the ITS domain. This version was available in the end of 2019, and it was used as basis to design and propose the recommender system that is an important contribution of this work and will be described later.
- **Version 1.1:** After the recommender system was mature enough, we refocused our attention to update the taxonomy and its knowledge base with articles describing projects released after Version 1.0. We included 12 new projects, including the three commercial applications (Waze, Uber and Moovit), and a simple rework in the categories was performed, to join some categories that have been shown during this period to be not so relevant as separate categories. One such example are the three Person UID categories (Person UID Knowledge, Person UID Physical Attribute and Person UID Possession) that were part of the taxonomy's Version 1.0. A review in the taxonomy in version 1.1 showed that they would be more useful and understandable as a single category (Person UID), with the previous categories used as examples of elements of each of these categories (these examples can be seen in Appendix C).

Extending the Knowledge-base is straightforward, given that a new ITS project is available and its contextual elements are well known. Adding a new project resumes to mapping the contextual elements used in the project to their respective categories in the taxonomy and append a new line for the project in the knowledge base, with its reference, and 0s for the columns of categories not used and 1s in the columns of categories that are used. Software houses or car makers with a vast history of projects could use an enhanced private version of the knowledge base with very little extra work. If such additions were allowed to be made public, we could add it in the official knowledge base of this research and deploy a new version in future works.

A relevant observation must be made regarding how new updates to the taxonomy can be made. Associating new contextual elements to existing categories is trivial, needing only an assessment of the currently available categories to make sure that the contextual elements are classified to the correct category(ies).

Creating new categories in the taxonomy, however, has a higher cost. The process involves: reviewing the current categories in the taxonomy to check the need and relevance of creating the new category; reorganizing the current categories according to the new category; reclassifying existing contextual elements that might be more fit in the new category than in the preivous one where they were classified; updating the knowledge-base to reflect which projects make use of the new category; evaluating the new version of the taxonomy. Thus, it is a labor-intensive work to update the taxonomy with new categories.

3.6 DISCUSSION

Our model has been defined to summarize and create a comprehensive taxonomy of Contextual Elements contributing to support the development of vehicular applications. Using (DEY; ABOWD; SALBER, 2001) as a starting point, we used four basic context types to create a structure that can categorize such elements. These four basic context types are called *supra-categories* in this work. This effort resulted in a hierarchical model with 57 leaf categories (those with no child subcategories). In total, the model consists of 79 categories, including the four supra-categories and all intermediary categories between them and the leaf categories. As far as we could trace, this is the broadest number of contextual element categories documented in one single place, specially designed for the vehicular application domain.

Designing an application with the support of the proposed model was helpful, and some of the features emerged by observing the model and identifying possibilities related to the application's core idea. When comparing our model to the work of Kannan, Thangavelu e Kalivaradhan (2010), we could check that our proposed taxonomy is indeed more general, and some of the features of the application designed as a part of the validation of this model would not be able to be modeled using the more specific domain existent in (KANNAN; THANGAVELU; KALIVARADHAN, 2010).

However, we have identified a limitation of the hierarchical format. The choice of Identity, Location, Time, and Activity was consistent with other contextual models aiming at a more generic one. Nevertheless, in this scenario, another valid alternative would be to root the model using the categories Vehicle, Driver, Passengers, and Environment. A model with similar expressiveness can be created if this is considered. While not a focus of this research, this observation must be taken into account if a future work arises on evolving this model into an ontology, which would be capable of representing both relationships.

We believe there is room for more formal qualitative research to improve the validation of the semantic value of this taxonomy. It would be necessary to elicit which measurements could be used as evidence of the effectiveness and afterward collect and analyze them.

In summary, this model is efficient and comprehensive enough to help design a context-aware ITS application. Although only one application was developed for validation, we believe the model is generic enough to be helpful on applications of this vast domain of Intelligent Transportation Systems. This belief is based on the robust literature we used to identify the contextual elements and categorize them into this taxonomy.

4 RECOMMENDING CONTEXTUAL ELEMENT CATEGORIES FOR INTELLIGENT TRANSPORTATION SYSTEMS

The taxonomy proposed in the previous chapter is a step forward on organizing the information of context-awareness usage in ITS applications. It allows that understanding the possible scenarios of use of contextual elements in past applications is straightforward. Together with the knowledge base, it can be used to guide system designers on defining the contextual elements that their new application requires.

However, using the knowledge base manually can still be an error-prone task. While the taxonomy by itself helps on understanding the relationships and possibilities that context-awareness offers in ITS, most of the information is in the knowledge base. And given its nature, this information is vast and not so easily readable and interpretable. By considering these observations, in this research we have also developed a process to recommend Contextual Element categories in the domain of Intelligent Transportation Systems.

Revisiting the motivation scenario described in Section 1.1, the recommender engine described in this section can be used in multiple moments of the software design and development life cycle. The ideal moment for using such a system is while the project is still on the requirements elicitation phase. In this stage, no system design or code has yet been created, and changes are still not expensive to be performed. If the requirements engineer already has the knowledge that the application will be benefited from using context-awareness, it is possible to already elicit the requirements with context in mind. Contextual elements that will be required to define context in the application can already be elicited together with the requirements of the application. In this scenario, the recommender system could be used in an iterative and incremental manner. The requirements engineer would run the recommender system using the already-found contextual elements categories used as input, verifying the recommendations and guiding the process of discovery of new requirements or contextual elements based on the categories that were recommended.

If it was not possible to use the recommender system in the requirements elicitation phase, either for lack of knowledge of its existence by the requirements engineer or by lack of perception of the need to use context-aware features in the application during the elicitation phase, it is still possible to use the recommender system. The system engineers, already conscious that the application will be context-aware, can analyze the output of the requirements elicitation and identify contextual elements required by the application. Then, prior to starting the definition of the system architecture, the system engineers might use our recommender system to identify further potentially useful categories of contextual elements to the application. After an analysis of the recommendations, it would be possible to identify contextual elements useful to the application, and validate them afterwards with the requirements engineer and other stakeholders of the project.

Finally, the recommender system would be used also in a more extreme scenario, where an application is already designed and developed. In this case, it would provide useful information to guide the development of improvements on the current application. In such scenario, the use of the recommender system would start by any software development team member of the project inputting the contextual element categories currently in use on the project. The recommendations would then be analyzed in the sense of identifying which of them could be useful to the project. In this scenario, there is the advantage that, since the application already exists and is probably used, there can be already feedback from users which could help on the analysis of the recommended items. The most promising categories could then be further analyzed and be used to guide the definition of the improvements for future versions of the application.

4.1 METHODOLOGY

The Recommender Engine for Contextual Element Categories is the major contribution of this work, using the taxonomy as the source of the possible items that can be recommended by the system. The research process of the Recommender System required a different methodology than what was performed on the design of the Taxonomy. The activities considered to design, develop and evaluate the recommender system were organized separately from the research on the taxonomy and knowledge base.

The first phase on designing the recommender engine was a viability analysis, investigating the possibilities that the knwoledge base could provide with the data it contains, to check the viability of creating a recommender engine based solely on such data.

Then, we performed a literature review on recommender systems, which is described in detail in section 2.3. In addition to learning about how recommender systems work, we also investigated the process of creating such systems. The Recommender Canvas proposed by Capelleveen et al. (2019) was chosen to guide the overall design decisions we used when creating this recommender system. Defining the values for the canvas fields was then performed.

However, some of the initial choices were refined and changed during the design, implementation and evaluation of our recommender system. The final recommender canvas for this recommender system is shown in Figure 11.

With a general understanding of the characteristics of the recommender system that we would create, we then moved on to the implementation phase. After the implementation, we performed an evaluation, that is further described later in this chapter.

4.2 RECOMMENDER ENGINE

The proposed method for recommending contextual element categories can be summarized by the following steps:

1. gather and analyze existing ITS projects,
2. identify Contextual Element Categories (from the ones defined in the taxonomy) in use in each one of them,
3. measure the similarity between the collected projects, and
4. devise a way to use these inputs to, given another project using an initial set of contextual element categories, suggest other categories that may contain elements missing in that set to be added to the project so that it can be even more useful and enjoyable for its users.

The first step (1) used the articles that were surveyed during the literature review of the taxonomy (explained in Section 3.2) to create the knowledge base described in Section 3.5.3. As already mentioned in Section 3.5.4, it was not possible to use the latest version of the knowledge base in the creation of this recommender engine, since the iteration that led to its creation was performed after the research on the recommender was already underway. We used the version 1.0 of the taxonomy and knowledge base to design and propose the recommender system, and this version comprised 61 projects. We used this version since it was the stable version of the taxonomy at the moment that the work on the design of the recommender system started. Version 1.1, which is also presented in this thesis, evolved at the same time that the research and evaluation of the recommender system were ongoing, so it was not possible to use it. We designed the recommender system to work with a dynamic knowledge base. Therefore, it is possible to switch to using the most updated version of the taxonomy and knowledge-base. A future work of this research is to validate the results against the new version.

Four of those projects were considered inappropriate, since they used only a single contextual element category and could not provide useful information on the relationship of categories that our type of recommender requires to work properly. For this reason, they were not used in the knowledge base of the Recommender System. Therefore, we ended up with a final number of 57 projects used to populate our knowledge base.

The knowledge base, consisting of a $Projects \times Context\, Element\, categories$ matrix was used to retrieve the relation of which contextual element categories are used in each project – the full matrix is available in Appendix A, which is the state of the version 1.0 of the knowledge base and taxonomy, without the four projects that are unfit for use in the recommender engine for using elements from only one category. It is a binary matrix of rows of projects and columns of contextual element categories, where value 1 in the intersection between a project and a category means that the project uses at least one contextual element of that category, and value `0` means otherwise. The articles describing the projects were thoroughly analyzed to identify which contextual elements were part of the associated ITS. Usually, there is a specific section providing the contextual information used to infer context in projects of applications that are explicitly context-aware. When the projects do not explicitly mention that make use of context awareness, it is necessary to gather information scattered in multiple sections of the articles, which requires more attention. In this case, some of the projects may use more contextual elements than it was possible to extract from the paper.

The aforementioned matrix was then used as input to calculate the *cosine similarity* of the projects. R was used as the programming language and environment to handle the input data, with the support of the `lsa` package – Latent Semantic Analysis (LSA) – to calculate the cosine similarity after the data have been properly parsed. The library uses each row of the matrix as a project, and each column as a dimension of the vector. It calculates the cosine similarity, according to Equation 2.4, to every possible pair of projects in the matrix.

The cosine similarity matrix for the 57 projects regarding the contextual element categories they use is available in Appendix B. A small sample of this 57×57 matrix is illustrated in Frame 6. The value in each cell of the table is the cosine similarity of the projects in the row and column of that cell. The higher the value, the more similar the projects are according to this metric. The cells where the project in the column and in the row are the same will always have the value 1, since it is the cosine between two equal vectors. A matrix similar to this

Figure 11 – Recommender System for Contextual Elements Categories Canvas

Recommender System for Contextual Elements Categories Canvas

Domain Characteristics

Roles of system users:
Requirements engineers on the process of
(1) Eliciting requirements for a new Context-Aware Intelligent Transportation System
(2) Improving an existing Context-Aware Intelligent Transportation System

Type of Available Data:
A binary matrix with the relations between Intelligent Transportation System projects and the Contextual Element Categories used in them. The matrix cells have a value 1 when the project in a cell's line uses a context element of the category of a cell's column, and 0 otherwise. The data in this matrix can be considered sparse, given the relation between the number of 1s and 0s.

Preference:
Preference might be a concern for this system, but it needs to be further investigated. Initial analysis showed no clear trend in the number of categories changing overtime, but we still need to understand whether there is a trend in the particular categories that are used in the projects over the time they were designed. Further investigation is now out of the scope.

Goals

Recommender goals:
- Help requirements engineers on identifying context element categories potentially missing in the design of their new ITS
- Provide assertive and accurate recommendations
- Ease the performed by the requirements engineer on the viability of using the recommended categories

Recommender use-cases:
John is a requirements engineer eliciting the requirements for a new ITS. He already performed requirements elicitation and identified some context elements that must be observed and used in the application. John has access to limited information regarding other similar projects which he could analyze to identify other elements and improve his application. With the help of this recommender system, John could receive suggestions based on a large number of projects, much more quickly than if he had to manually analyze the projects.

Functional Design Considerations

Degree of personalization:
Initially, no degree of personalization is needed. A future iteration could allow changing the similarity metric algorithm or update the knowledge base with information from the user's or its employer's private projects.

Degree of user control:
The user has control of the base project context element categories so the recommender system is able to process it against its knowledge base.

Interactivity:
Simple interaction of inputting the categories of the base project and receiving the result of the engine.

Technique Selection

Filtering algorithm:
Knowledge-based filtering is most recommended for the purposes of this recommender system. The use of information from the knowledge base, which already contains a representation of the domain, is one of the reasons for this choice. Its resilience against the cold-start problem is also interesting.

Dimensionality reduction & Scalability:
Given that the knowledge base used in this recommender system is manually built, it is not expected that the underlying data used as source for this recommender system will grow exponentially, with an expectation that a large base will contain at most a few hundreds of projects. Under these numbers, there is no need for dimensional

Evaluation & Optimization

Evaluation:
- Use accuracy-based metrics to evaluate the results.
- Create a process to use the information available in the knowledge base to measure the accuracy of the tool
- Verify that the number of recommendations is a good compromise between the goals of easing the analysis, but still providing assertive and accurate recommendations.

Optimization:
No optimization efforts are expected in this first iteration of the recommender system.

Protection:
The single-user model and the currently read-only state of the knowledge-base make protection a non-concern for this iteration of the recommender system.

Interface design

Presentation modality:
Visual and text only, with a GUI where the user will provide the categories as input and will receive the list of recommendations.

Item organization:
Recommended items must be shown as a list, ordered by their relevance according to the user input. Showing a value that indicates this relevance is also interesting.

Item notification:
Items are only recommended after the user confirms having finished providing the input.

Item information:
No extra information is expected in this iteration. However, future iterations could provide hyperlinks from the recommended categories to examples of its context elements and of projects that use them.

Item explanation:
The system must provide to the users the projects that were considered similar to the base project. As such, the user is able to read more about the projects and have more data to decide whether using elements from the suggested category would really be a good fit to their new projects.

Source: The author (2022)

one is calculated by the recommender engine every time the user inputs a new base project. The main difference is that instead of 57 projects, the matrix calculated by the recommender engine will have 58 projects: the base project with the categories informed by the user in the Graphical User Interface (GUI) of the application, plus the 57 projects in the knowledge base.

Frame 6 – Sample of the cosine similarity matrix.

	(SILVA; BORGES; VIEIRA, 2018)	(VIEIRA; CALDAS; SALGADO, 2011)	(KANNAN; THANGAVELU; KALIVARADHAN, 2010)	(MEIER; HARRINGTON; CAHILL, 2006)	(AGUIRRE et al., 2017)	(YOUNES; BOUKERCHE; MAMMERI, 2016)	(SUKODE; GITE, 2015)	...	(GOTO; KAMBAYASHI, 2002)
da Silva et al. (2018)	1,000	0,183	0,320	0,000	0,183	0,000	0,000	...	0,000
Vieira et al. (2011)	0,183	1,000	0,175	0,730	0,300	0,316	0,183	...	0,224
Kannan et al. (2010)	0,320	0,175	1,000	0,160	0,526	0,277	0,160	...	0,000
Meier et al. (2006)	0,000	0,730	0,160	1,000	0,274	0,289	0,333	...	0,510
Aguirre et al. (2017)	0,183	0,300	0,526	0,274	1,000	0,316	0,183	...	0,000
Younes et al. (2016)	0,000	0,316	0,277	0,289	0,316	1,000	0,577	...	0,000
Sukode and Gite (2015)	0,000	0,183	0,160	0,333	0,183	0,577	1,000	...	0,000
...	...	...	...	...	...	...	...	...	...
Goto and Kambayashi (2002)	0,000	0,224	0,000	0,510	0,000	0,000	0,000	...	1,000

Source: The author (2022)

To get the recommendation of potentially useful categories in a project, a user must provide the tool a set of contextual element categories used in their project. This project is referred to in this document as the *baseline project* and its set of categories as the *baseline contextual element categories*. The recommender engine recomputes the similarity matrix on every execution. The similarity values are then used to compare the baseline project with the projects in the knowledge base.

The cosine similarity indexes are valuable to understand which projects share similar usage of contextual elements. Based on this similarity index, it is possible to use a Top-n approach that considers the n projects with higher similarity indexes. Part of the validation done in this research aimed on identifying the best values for n (the cut-line for projects whose contextual elements categories can be effectively valuable for the new project being designed). Then, we compare the set of contextual elements categories used in each of the projects with the categories used in the baseline project. The objective is to identify which categories are present in the similar projects but are not used in the baseline project. However, this can lead to a vast list of categories, and we would need a way to order these categories to better guide the person in charge of analyzing whether a suggested category would be a good fit to the baseline project.

To order the suggested categories in a meaningful way, we define a *relevance index* as the sum of the similarity indexes of the n projects where contextual elements categories are used, divided by n. Thus, the *relevance index* will be a value between 0 and 1, which takes into account:

a) How similar the project is in relation to the other project where a contextual element category is also used; and

b) In how many of the similar projects that category is used.

This *relevance index* is the utility function used in the recommender system to hypothesize how relevant a recommended category will be to the user.

The process to generate an ordered *suggestion list* of contextual elements categories based on the known projects using this recommender engine can be formalized as follows:

1. There is a set P of projects: $P = \{p_1, p_2, ..., p_k\}$;
2. Each of these k projects has a set E_k of contextual element categories: $E_k = \{e_1, e_2, ..., e_j\}$;
3. A set Eb of baseline contextual element categories is provided to the recommender engine;
4. The recommender engine creates a baseline project pb with Eb as its set of contextual element categories;
5. A set P' is created by adding pb to P: $P' = P \cup \{pb\}$;
6. The cosine similarity matrix for P' is calculated;
   - The cosine similarity index between the project pb and a project p_x is Cb_x;
7. The Top-n projects in set P with the highest similarity index values (i.e the n most similar projects to pb) are selected into set Pn: $Pn = \{pn_1, pn_2, ..., pn_n\}$;

8. A set S of suggested contextual element categories is created empty ($S = \{\emptyset\}$);

9. The set S is populated with the categories used in the Pn's projects that are not used in pb: $\forall p_n \in Pn, S = S \cup (E_n - Eb)$;

10. For each category of E_x in S, its *relevance index* i_x is defined as in Equation 4.1:

$$i_x = \frac{\sum_{y=1}^{|Pn|} \begin{cases} Cb_y & , if\ E_x \in Pn_y \\ 0 & , otherwise \end{cases}}{n} \tag{4.1}$$

11. Finally, order S by the relevance index i_x of each of its category in E_x.

Example

To further explain the process through an example, we will use a small part of Frame 6. Instead of taking the complete set of 57 projects into account, we will consider the first **six projects** in Frame 6 (SILVA; BORGES; VIEIRA, 2018; VIEIRA; CALDAS; SALGADO, 2011; KANNAN; THANGAVELU; KALIVARADHAN, 2010; MEIER; HARRINGTON; CAHILL, 2006; AGUIRRE et al., 2017; YOUNES; BOUKERCHE; MAMMERI, 2016) as the *knowledge base*, and the **seventh project** (SUKODE; GITE, 2015) as the *baseline project* – note that in real-world usage of the recommender system, instead of considering an existing project, the new project under development would be the *baseline* one, with the known used categories to be provided to the recommender system.

Let $n = 2$, so the two most similar (Top-2) projects to the baseline project (SUKODE; GITE, 2015) are: (YOUNES; BOUKERCHE; MAMMERI, 2016), here called topN-1, and (MEIER; HARRINGTON; CAHILL, 2006), called topN-2, with *similarity indexes* of 0.577 and 0.333, respectively.

To generate the recommendation lists, it is thus important to check the categories listed in the baseline project and in the Top-2 similar projects:

- $Baseline\ project$ = {*Traffic Condition*; *Surroundings Weather*; *Surroundings Traffic Lights*}

- $topN\text{-}1$ = {*Traffic Condition*; *Surroundings Traffic Lights*; *Vehicle Activity*; *Vehicle Type*}

- $topN\text{-}2 = \{$*Traffic Condition*; *Surroundings Weather*; *Vehicle Id*; *Vehicle Type*; *Place Id*; *Geographic Coordinate*; *Geographic Location*; *Location Type*; *Route*; *Time*; *Schedule*$\}$

Without considering the *relevance index*, the recommendation result would be a set with all the categories used either in topN-1 or in topN-2 that are **not** part of the *baseline project*, so

$$\begin{aligned} S = \{ & \textit{Vehicle Activity}; \textit{Vehicle Type}; \textit{Vehicle Id}; \textit{Place Id}; \textit{Geographic} \\ & \textit{Coordinate}; \textit{Geographic Location}; \textit{Location Type}; \textit{Route}; \textit{Time}; \textit{Schedule}\} \end{aligned}$$

Calculating the *relevance indexes* (see Eq. (4.1), page 96) for each of the recommended categories in S helps to order the results based on the number of the top-n projects where each category is present. For the computation of the relevance index, the projects' similarity indexes related to the baseline project are also taken into account. For instance, the *relevance index* for the category *Vehicle Type* is the highest in the S set as it is used in both projects. Its value is calculated as $(0.577 + 0.333)/2 = 0.455$. Even though the *Vehicle Activity* category is only used in the topN-1 project, its *relevance index* is the second-highest due to topN-1's similarity index being higher than that of topN-2 project. It is calculated as $(0.577+0)/2 = 0.289$. Since the remaining categories in S are only used in the topN-2 project, their *relevance indexes* are equally calculated as $(0 + 0.333)/2 = 0.167$, and thus, they are positioned at the end of the suggestion list. The final S list, ordered from highest to lowest *relevance index*, is

$$\begin{aligned} S_{list} = [& (\textit{Vehicle Type}, 0.455); (\textit{Vehicle Activity}, 0.289); (\textit{Vehicle Id}, 0.167); \\ & (\textit{Place Id}, 0.167); (\textit{Geographic Coordinate}, 0.167); \\ & (\textit{Geographic Location}, 0.167); (\textit{Location Type}, 0.167); (\textit{Route}, 0.167); \\ & (\textit{Time}, 0.167); (\textit{Schedule}, 0.167)] \end{aligned}$$

Although such a large number of ties concerning the relevance index is not desirable, higher values of n tend to prevent this from happening. We have not defined any tiebreaker criteria because the relevance of all tied elements is absolutely the same. That implies that, if there was a limit to avoid information overload for the user (say, $limit = 6$), exceptionally, the S_{list}

should have all the elements whose relevance indexes are equal to the sixth element. This way, in the example above, without such an exception, the list would end in the sixth element (*Geographic Location*, 0.167), but as it can be seen, the list contains all ten elements, since the remaining four (Location Type; Route; Time; Schedule) have the same *relevance indexes* = 0.167.

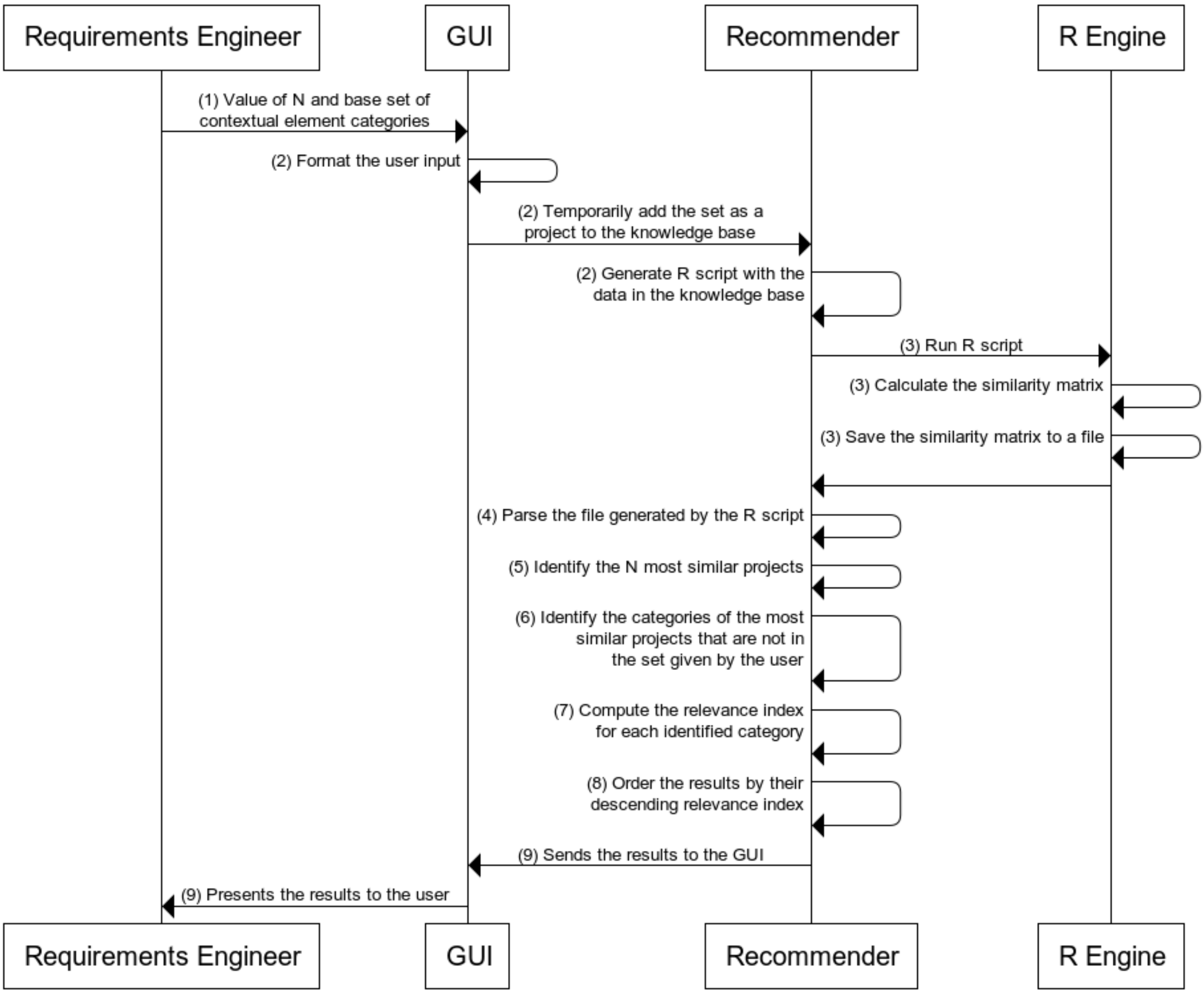
Finally, the descending-ordered S_{list} should help the requirements engineer to decide which recommended contextual element categories to incorporate for an enhanced context-aware project. The Recommender process flow is described in Figure 12, and can be further explained as follows:

1. A requirements engineers (user) **inputs** the contextual element categories known as needed to the project being designed (p_x);
2. A C# library formats the user input and appends it in the knowledge base;
3. An R script generates the similarity matrix considering the data in the knowledge base;
4. The C# library parses the matrix generated by the R script;
5. Identify the n projects whose categories show a higher similarity to the set of categories given as input;
6. In these n projects with the highest similarity to p_x, identify the categories that are not part of the user's input;
7. Compute the relevance index i_x for each of the identified categories;
8. Arrange the categories by their relevance index i_x in descending order;
9. Return the ordered list of categories and their respective relevance indexes to the user. This **output** is then the *list of recommended categories*, so the user can decide (taking the relevance indexes into account) which new categories will be included in the context-aware vehicular application under development.

Implementation

The process above has been implemented using a mix of C# and R. Source code 1 contains the R script that is generated in step 2 and executed in step 3 to calculate the similarity matrix

Figure 12 – Recommender process



Source: The author (2022)

of the items in the knowledge base plus an hypothetical project that uses the categories informed by the user in step 1, and is represented in code by the variable $proj0$.

Source Code 1 – R script generated by the C# application to recalculate the similarities with the baseline project

```
library(lsa)

proj0 = c(1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
          0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,0,1,
          1,1)

proj01 =c(1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
          0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,
          0,0)

# ...
# Here goes the binary representation of the lines of the other projects,
```

```
# except projects 4, 9, 37 and 50, which use only one context element category.
# ...

proj61 =c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,
          0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,1,0,0,0,1,
          1,0)

bindProjs = cbind(proj0, proj01, proj02, proj03, proj05, proj06, proj07,
                  proj08, proj10, proj11, proj12, proj13, proj14, proj15,
                  proj16, proj17, proj18, proj19, proj20, proj21, proj22,
                  proj23, proj24, proj25, proj26, proj27, proj28, proj29,
                  proj30, proj31, proj32, proj33, proj34, proj35, proj36,
                  proj38, proj39, proj40, proj41, proj42, proj43, proj44,
                  proj45, proj46, proj47, proj48, proj49, proj51, proj52,
                  proj53, proj54, proj55, proj56, proj57, proj58, proj59,
                  proj60, proj61)

result = cosine(bindProjs)

write.csv(result,"tempSimResult.csv")
```

Source: The author (2022)

The code presented in Source code 2 retrieves the top-n most similar projects according to the cosine similarity calculated with the R script. For illustrative purposes, the n value is fixed to 5. Line 3 calls a method that parses the csv result obtained from the script execution (step 4 of the process flow). The code for this parsing is utilitarian and is omitted here. Step 5 is implemented by the GetTopNSimilarItems method that makes use of the LINQ capabilities of C# to select the the top-n projects with the greater value of the cosine similarity, which is the value returned by the lines 13-16 of the following code.

Source Code 2 – Obtaining the n most similar projects in C# based on the cosine similarity returned from the R script execution

```
   //CERecommenderMain.cs:
 2
   ParseData data = ParseData.CreateInstance(outputFile, "tempSimResult.csv");
 4 int n = 5;
   var topN = Similarity.GetTopNSimilarItems(data.Similarities, "proj0", n)
 6                      .ToList();

 8 // Similarity.cs:

10 public static IEnumerable<Similarity> GetTopNSimilarItems(
                                        List<Similarity> similarities,
12                                      string item, int n) =>
```

```
    similarities.Where(x => x.ProjectA == item
                        && x.ProjectB != item)
                .OrderByDescending(x => x.CosineIndex)
                .Take(n);
```

Source: The author (2022)

Source code 3 contains the code for the FindPotentiallyMissedCategories method, which is the implementation of the step 6 of the process. This method identifies all categories that were used in the top-n most similar projects, but are not in the baseline project provided by the user.

Source Code 3 – Identification of the potential recommendation results

```
public static Dictionary<int, double> FindPotentiallyMissedCategories(
                                        Dictionary<string, bool[]> elements,
                                        string baseProj,
                                        int n,
                                        List<Similarity> topN)
{
    var topNelementsBase = elements[baseProj];
    var topNelementsAcc = new double[elements[baseProj].Length];

    foreach (var item in topN)
    {
        var itemElements = elements[item.ProjectB];

        for (int i = 0; i < itemElements.Length; i++)
        {
            if (!topNelementsBase[i] && itemElements[i])
            {
                topNelementsAcc[i] += item.CosineIndex;
            }
        }
    }

    Dictionary<int, double> result = new Dictionary<int, double>();
    for (int i = 0; i < topNelementsAcc.Length; i++)
    {
        if (topNelementsAcc[i] != 0)
        {
            result.Add(i, topNelementsAcc[i] / n);
        }
    }

    return result;
}
```

Source: The author (2022)

A performance improvement of the process was done during the implementation, and can also be seen in Source code 3. Lines 16-19 and 26-29 are performing the step 7 of the process, the computation of the relevance index. This was performed together with the code that implements the step 6 so the list must not be iterated all over again. Code for the steps 8 and 9 is trivial and will be omitted in this manuscript.

4.2.1 Contextual Elements Category Recommender Tool

A GUI application was developed to allow vehicular application designers to easily have access to the features of the proposed recommender system. We focused on the simplicity of this application so the user is not hassled with unnecessary complexity.

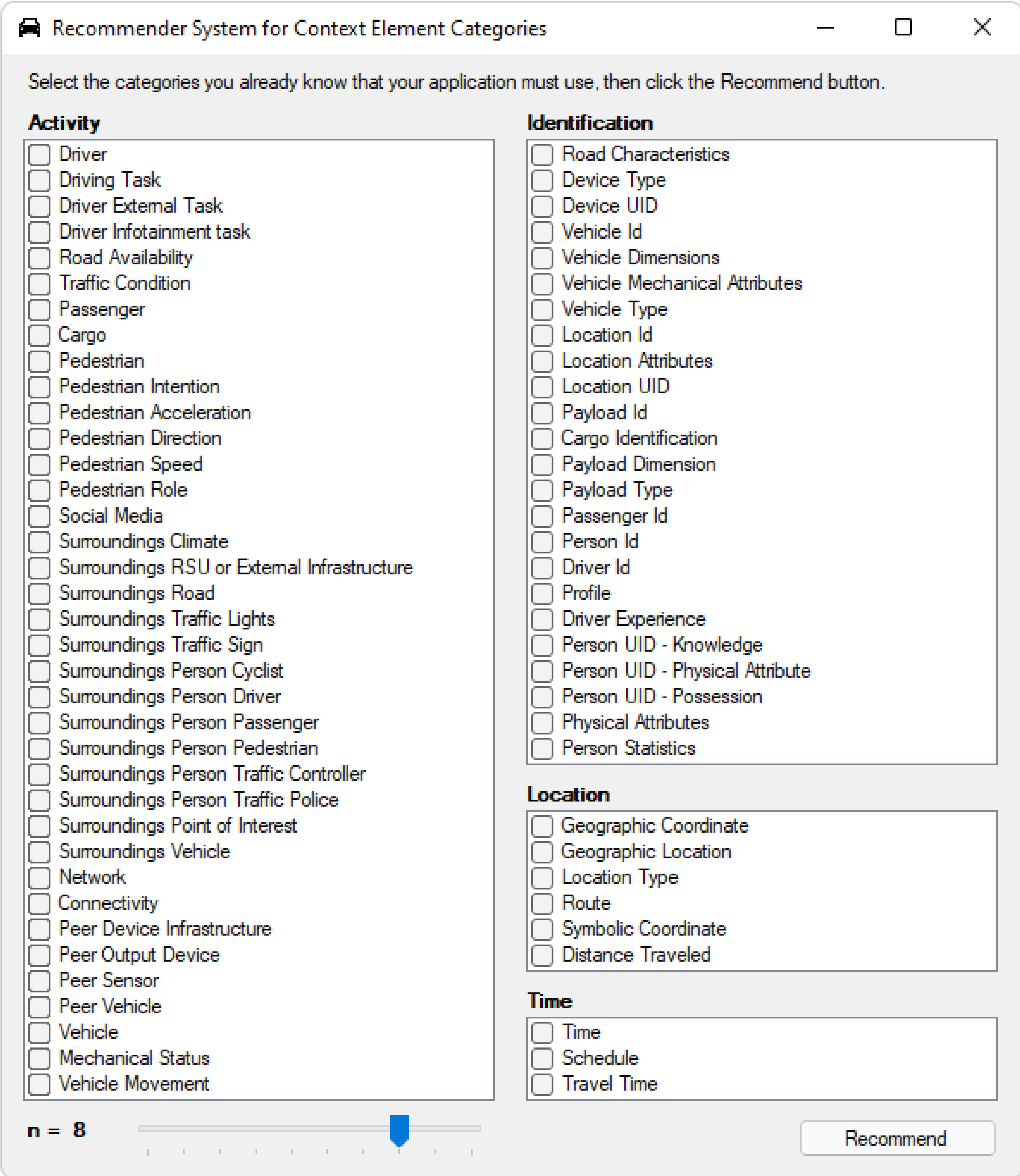
The goal of a user of this software is to obtain the recommendations, given an initial set of contextual element categories that are already known to be required in the application being designed by the user. To achieve this goal, the user must check the boxes corresponding to the known categories in the tool's first screen, shown in Figure 13. It is also possible to set up the desired value of n, used to select the top-n most similar projects to the base line. The value can be in a range between 1 and 10, and comes with the default value of $n = 8$ – a study to define 8 as the default value is discussed in Section 4.4.

After selecting the known categories, the user can then click on the Recommend button, where the engine will calculate the similarity matrix of the knowledge base plus the given baseline project created with the informed categories, and generate the recommendation result, as shown in Figure 14. The list is ordered descendingly by the relevance index of each recommended category. This allows a quick identification by the user of which are potentially the best categories to be investigated.

The result dialog provides an option to the user, after checking the recommendations: Seeing which were the most similar projects to the provided baseline. When the user clicks on the *Yes* button, a screen similar to Figure 15 is shown, with the top-N projects used in the recommender engine to generate the list of the recommended contextual element categories. The user can then search the articles and read further about the projects that were identified as most similar to the one being designed.

The user can also vary the value of n, as previously mentioned, to instruct the recommender system to use a value between 1 and 10 most-similar projects to the baseline project provided

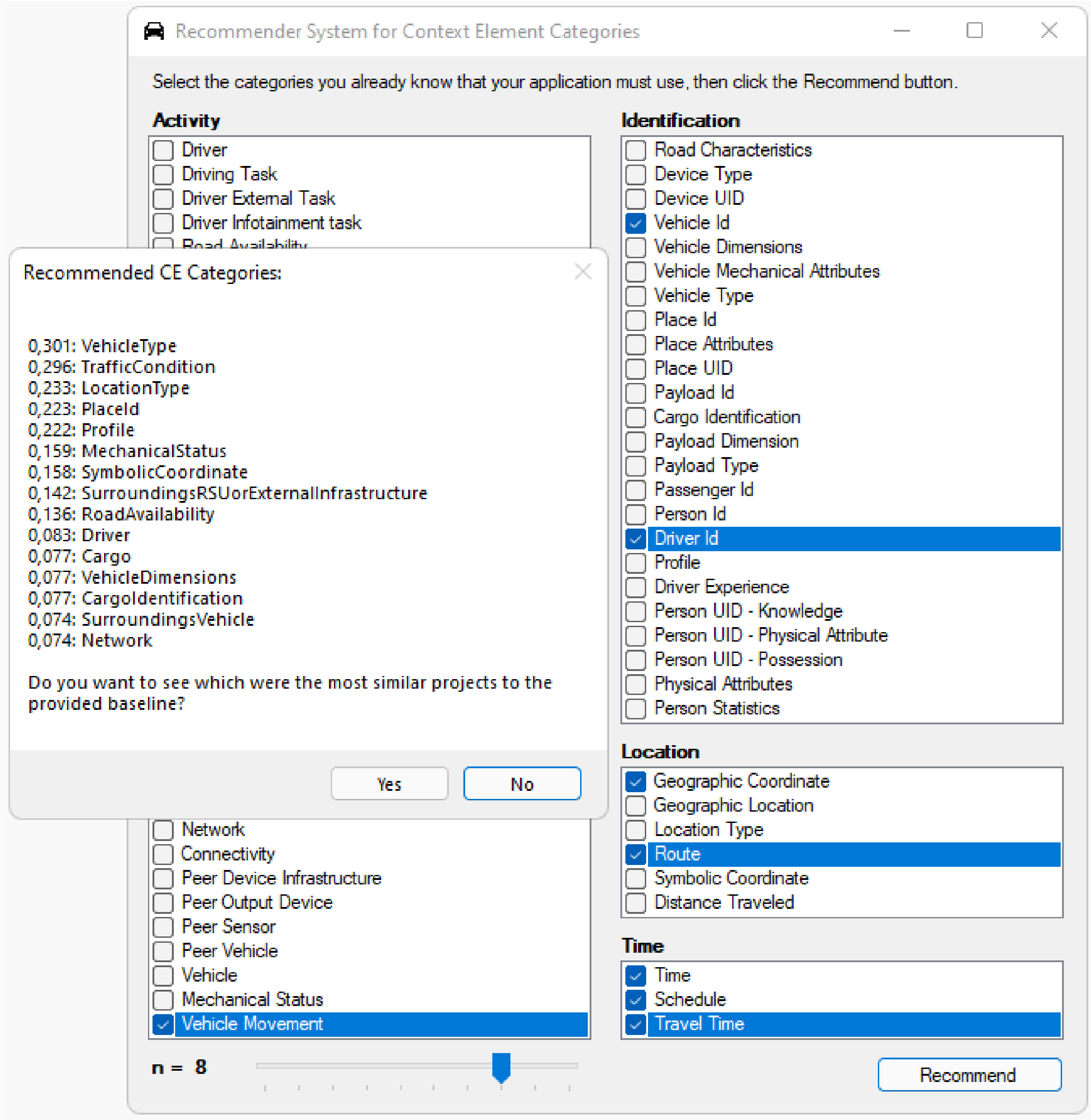
Figure 13 – Recommender tool - Initial screen



Source: The author (2022)

as input. Figure 16 shows the result of running the recommender system on the same baseline project provided in Figures 14 and 15, however increasing the value of n from 8 to 10. The variations in the response that happened due to this change are highlighted in red. In the bottom of the screen, it is possible to check that the user has increased the value to 10 on the slider bar. The dialog to the left of the screen shows that the list of recommended categories

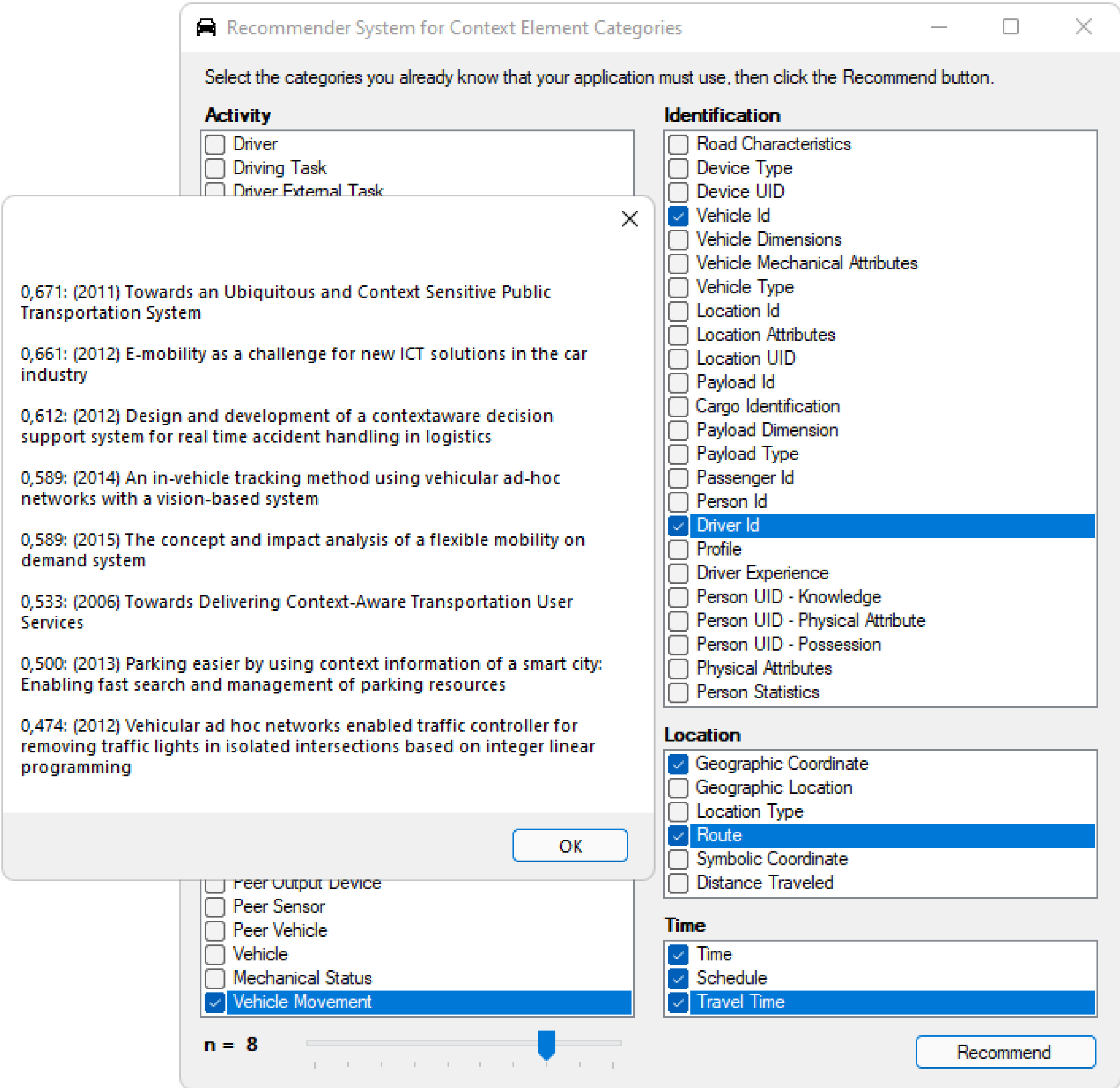
Figure 14 – Recommender tool - Result



Source: The author (2022)

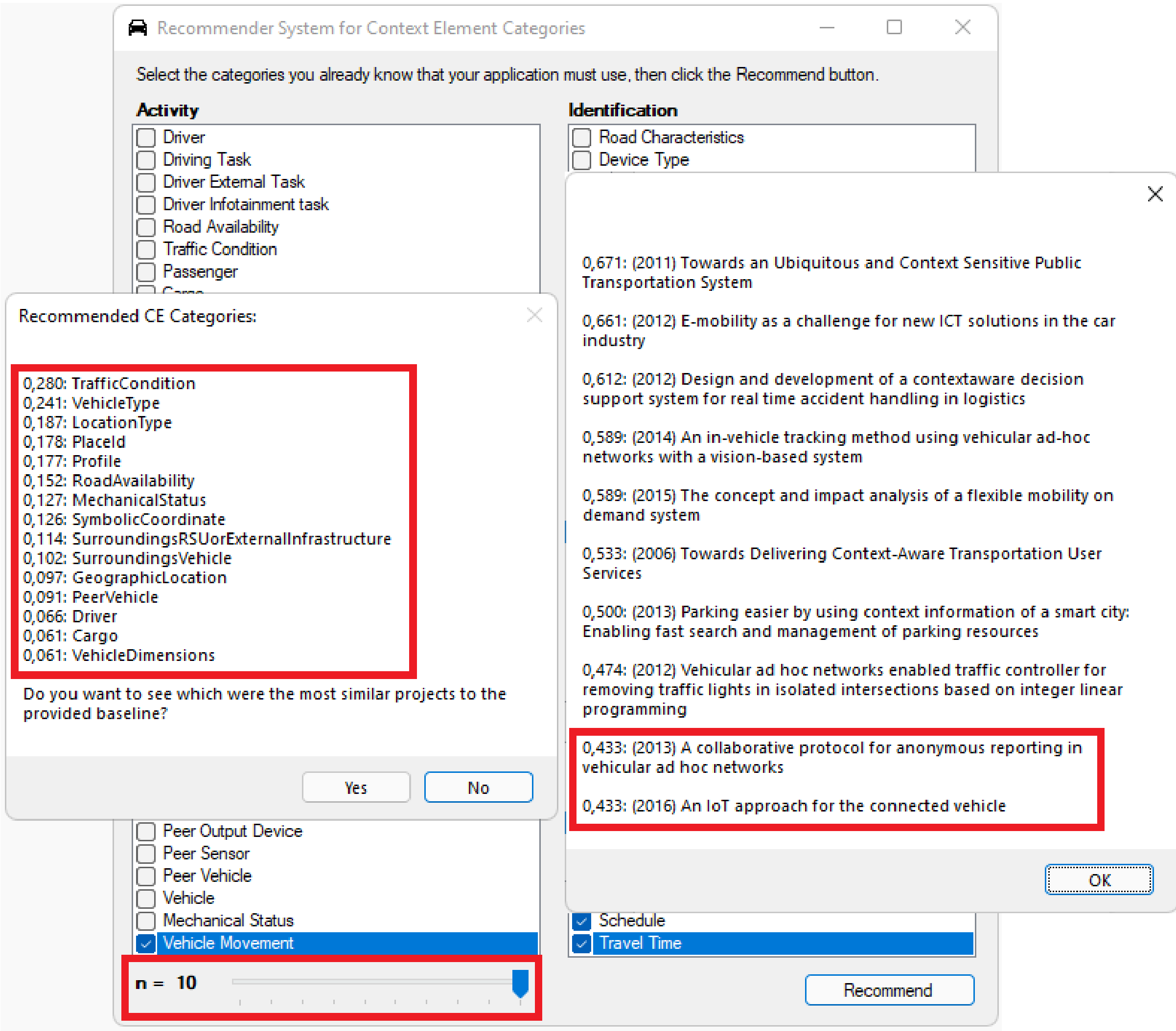
has a slight variation, with some changes in the ranking and some of the less-relevant items provided when using $n = 8$ giving way to other categories that had a higher relevance index when $n = 10$ was used. The two projects that were elected as similar when increasing n from 8 to 10 are also highlighted in the dialog to the right of the screen.

Figure 15 – Recommender tool - Similar projects



Source: The author (2022)

Figure 16 – Recommender tool - Varying n-value for the same set of baseline contextual element categories



Source: The author (2022)

4.3 EVALUATION

To evaluate whether the methodology would provide significant results, we chose some of the original 57 projects analyzed to create the previously mentioned matrix as the baseline for comparison. In this first, more informal evaluation, the projects were chosen based on the understanding of their domains. This step was performed as an empiric observation of the recommendations to rule out the possibility of the process outputting nonsensical results. We ran the process 57 times using each of these projects as a *baseline project*, and checked if the suggested categories were compatible with those of the same project.

Overall, the suggestions received during this evaluation were satisfactory. For instance, a project in the knowledge base has the goal of finding and reserving parking spaces. It indeed makes use of several context element categories, such as vehicle, driver and place identification, locations attributes (such as for parking lots and individual parking spaces), driver profile info, geographic coordinates and locations, among others. For this given set, some of the suggested categories were location type (0.181 *relevance index*), schedule and time (*relevance indexes* of 0.131 and 0.127, respectively). These categories are very probably useful for this project. Indeed, the schedule category might even be already used, however, since the description in the project article is not specific about its usage, we did not include it in the project's list of used categories, since false-positives are much worse for our recommender system than false-negatives, as explained previously in Section 3.5.3.

We also observed this happening with several other projects, where we are almost sure that the project uses some of the recommended categories, but we opted for not including these in the knowledge base due to the manuscripts about the projects not being clear and specific regarding the use of contextual elements from these categories. One such example is a system to automatically warn authorities and health services of car crashes that was not explicit regarding the use of the geographic coordinates (recommended as the most relevant category for its case, with a *relevance index* of 0.303). It is highly unlikely that such a system does not already use the geographic coordinates category, but as the article is unclear about this, we have not included it in the list. The recommender system, however, noticed the pattern and suggested these categories, along with others that do not seem to be used currently in the system but could really provide important information to the application's objectives, such as Road Characteristics, Traffic Condition and Cargo.

After that initial evaluation has provided insights that the suggestions made based on the proposed method were adequate, we formalized an *evaluation process* based on the projects that had their contextual elements mapped. This *evaluation process*, illustrated in Figure 17, considers that:

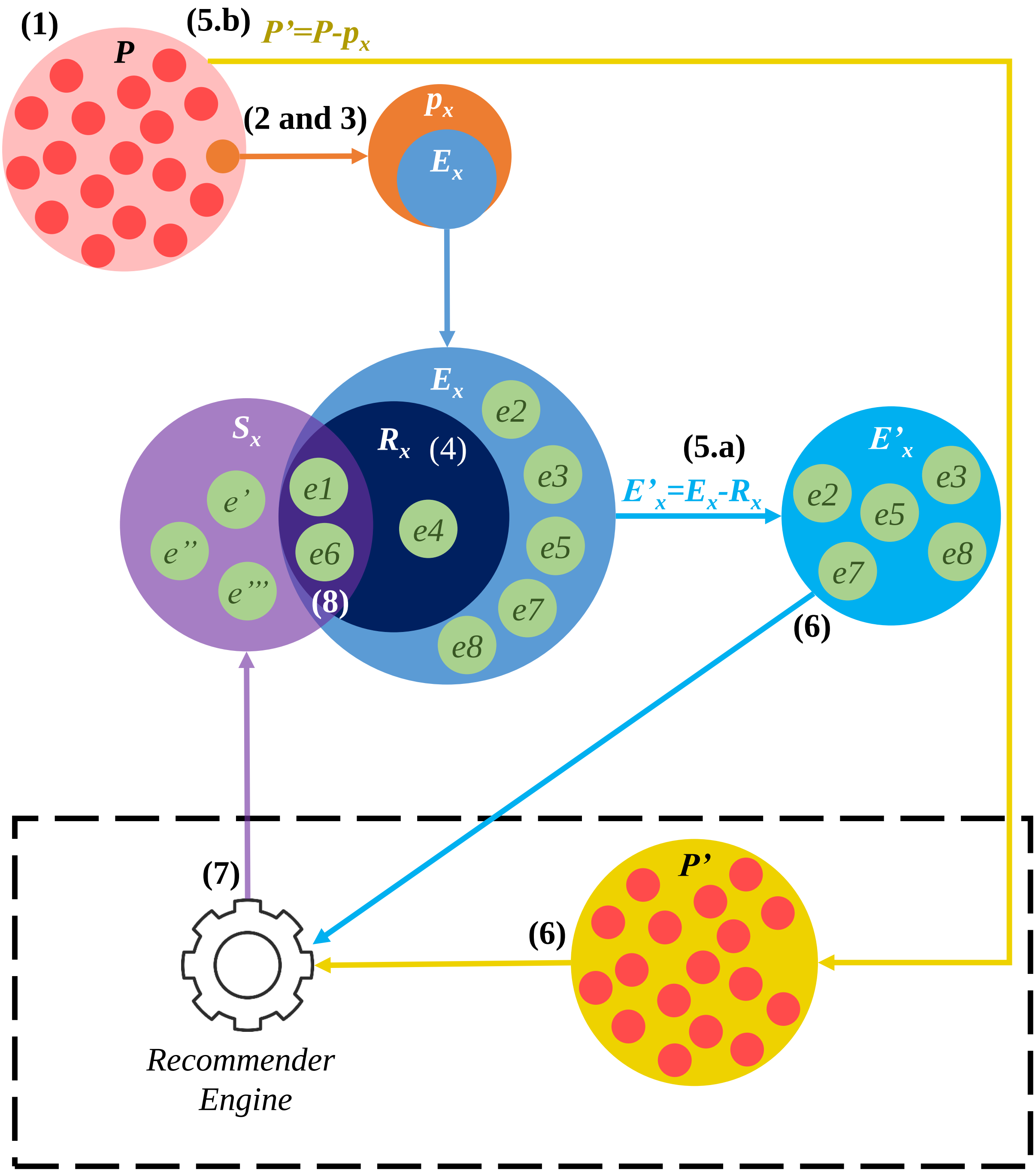
1. There is a set P of projects: $P = \{p_1, p_2, ..., p_k\}$;
2. Each of these k projects has a set E_k of contextual element categories: $E_k = \{e_1, ..., e_j\}$;
3. We select a project p_x in P and remove it from P: $P' = P - \{p_x\}$; p_x is now considered a *baseline project*;
4. We randomly select some categories in the E_x set to create a non-empty subset R_x of removed categories from E_x;
5. a. We define $E'_x = E_x - R_x$;

   b. We define $P' = P - \{p_x\}$;
6. We run the Recommender Engine on P', providing E'_x as the set of baseline contextual element categories;
7. The recommender engine returns the set S_x of suggested contextual element categories;
8. One expects $R_x \subseteq S_x$. If $|R_x \cap S_x| = |R_x|$, then the result is **wholly successful**. Else, if at least $0 < |R_x \cap S_x| < |R_x|$, then it is **partially successful**.

Example

Figure 17 shows that, given a set P of k projects and their respective sets of contextual element categories $E^k_{i=1}$, an evaluation starts by selecting a project p_x from P to be the *baseline project* of the experiment (**step 3** in the previously formalized process). Then, some elements are randomly selected to be removed from p_x's set of contextual element categories, $E_x = \{e1, e2, e3, e4, e5, e6, e7, e8\}$, to create a subset $R_x = \{e1, e4, e6\}$ (**step 4**). The elements that were not picked to be part of R_x form a new set $E'_x = \{e2, e3, e5, e7, e8\}$ (**step 5.a**). In **step 6**, E'_x and P' are provided as inputs to the Recommender Engine, which recommends contextual element categories making set $S_x = \{e1, e6, e', e'', e'''\}$ (**step 7**) – the execution of the recommender engine is described in detail further in this section. The

Figure 17 – Evaluation process flow.



Source: The author (2022)

evaluation then proceeds by checking the intersection between R_x and S_x (**step 8**), that is $\{e1, e4, e6\} \cap \{e1, e6, e', e'', e'''\} = \{e1, e6\}$. In this case, even though not all R_x's elements are in S_x, i.e., $|R_x \cap S_x| \neq 3$ ($|R_x|$), at least the criterion for **partial success** is met, that is $|R_x \cap S_x| > 0$.

This evaluation is repeated for every project in P and it relies on the premise that the projects are using carefully chosen contextual elements, which are meaningful to the projects' goals. The logic behind such an evaluation is that, given the previously mentioned premise, since project p_x originally had the categories in R_x, these are supposed to be related to the remaining contextual element categories in E'_x. In this case, we rely on the coherence of the elements used in the projects analyzed in this research.

While one expects that the engine can provide a suggestion list where $R_x \subseteq S_x$, meaning that S_x contains all the categories in R_x, it is subject to factors such as the cardinality of E'_x, which may be too small to yield valid results. Indeed, while not an objective of this evaluation, using it would be possible to identify a threshold below which there are too few contextual element categories in the set, so the engine cannot provide reliable results.

Besides identifying whether the recommender engine suggests back the removed element categories (R_x), we also need to consider the *number of suggested items* ($|S_x|$). Moreover, we want to evaluate the *relevance of the categories* in $R_x \cap S_x$, which is done by considering the positions where they appear in descending order by relevance index in the suggestion list.

The experiments were designed to be run varying two factors: the N value (to select the top-n similar projects) and the cardinality of R_x (meaning the number of elements removed from E_x). We ran an experiment for $1 \leq n \leq 10$, and for each value of n, we executed it varying the cardinality of R_x such that $1 \leq |R_x| \leq 5$, but when a project has less than 5 categories, $|R_x|$ varies up to the maximum number of $|E_x| - 1$ (12 of the 57 projects of the experiment have less than 5 categories).

For each of these n and $|R_x|$ combinations, for each project p_x in P, we ran 10 random selections for the categories to compose R_x (based on the E_x set of p_x) and executed the recommender engine for each of these values. Duplicate runs, which might happen given that the random selection of categories to R_x can lead to repeated E'_x sets being generated, were identified and only one of the runs was considered in the results analysis. Since the recommender engine is deterministic, the result of any such duplicated run would be equal to the other duplicated runs of the same E'_x set.

Source code 4 contains the RunAllProjects method, that is used by the code implemented to perform this evaluation to run the process for each n and $|R_x|$ combination. It performs all calculations for all projects to the given value of n and $|R_x|$. The $|R_x|$ is represented in the code by the *categoriesToRemove* method argument.

Source Code 4 – Code used to execute the experiment for a given value of n and r.

```
private static List<string> RunAllProjects(int categoriesToRemove,
```

```
                                              int n,
                                              ParseData rawData)
{
    List<string> runs = new List<string>();

    foreach (var project in rawData.Header)
    {
        Console.WriteLine(project);
        var indexesToFlip = new List<int>();
        var allCategories = rawData.Categories[project];
        var originalAllCategories = new bool[allCategories.Length];
        Array.Copy(allCategories, originalAllCategories, allCategories.Length);

        var usedCategories = allCategories.Select((x, y) => new { Index = y,
                                                          HasCategory = x })
                                          .Where(x => x.HasCategory)
                                          .Select(x => x.Index)
                                          .ToList();

        var maxCategoriesToRemove = Math.Min(categoriesToRemove,
                                             usedCategories.Count() - 1);

        for (int i = 0; i < maxCategoriesToRemove; i++)
        {
            var topLimit = maxCategoriesToRemove - i;

            var category = usedCategories[random.Next(usedCategories.Count())];
            indexesToFlip.Add(category);
            usedCategories.Remove(category);
        }

        foreach (var item in indexesToFlip)
        {
            allCategories[item] = false;
        }

        var newData =
                $"{rawData.CategoriesToString()}\n{rawData.HeadersToString()}";
        var similarityFile = Path.Combine(Directory.GetCurrentDirectory(),
                             DateTime.Now.GetTimestampedFilename("sim", "csv"));
        var templateFile = RUtils.CreateTemplateFile(newData, similarityFile);

        RUtils.CallR(templateFile);

        var resultData = ParseData.CreateInstance(templateFile, similarityFile);
        var topN = Similarity.GetTopNSimilarItems(resultData.Similarities,
                                                  project, n)
```

```
49                              .ToList();
           var result = Similarity.FindPotentiallyMissedCategories(
51                                                         resultData.Categories,
                                                           project, n, topN);
53
           var resultContainsRemoved = result.Keys.Where(x =>
55                                                       indexesToFlip.Contains(x))
                                                 .Count();
57
           var str1 = $"{project};{resultContainsRemoved};";
59         var str2 = $"[{string.Join(",", indexesToFlip)}];";
           var list = result.OrderByDescending(x => x.Value)
61                            .Select(x => $"({x.Key},{x.Value * 100:00.00})");
           var str3 = $"[{string.Join(",", list)}]";
63
           runs.Add($"{str1}{str2}{str3}");
65
           Array.Copy(originalAllCategories, allCategories, allCategories.Length);
67     }

69     return runs;
   }
```

Source: The author (2022)

The *Similarity.cs* part of Source code 2 and all of the Source code 3 is also used by this process, in lines 47 and 50 respectively. Also, an R script equivalent to Source code 1 is generated in line 42, but without including a new line for the baseline $proj0$. Instead of this, the line of code representing the categories used by project of the current iteration of the `foreach` in line 7 is replaced with the list of categories in E'_x (which is generated in lines 15-31). Thus, step 6 of the evaluation process happens, because instead of using P as the knowledge base, the evaluator is using $P' + E'_x$, since the original line of code representing the categories used by project p_x has been replaced with the categories in E'_x.

4.4 RESULTS

After executing the experiments following the evaluation process described in the previous section, we obtained the recommended contextual element categories for 18,598 distinct combinations (or "runs") of n, $|R_x|$, and E'_x. For each group of runs with equal values of n and $|R_x|$, the number of distinct runs varied from 299 to 436. Since R_x's elements are selected randomly from the E_x set, in some cases ("runs") the same elements could be selected. A *distinct run* is when no other run has used the same combination of n and R_x. Since the process is deterministic, running the recommender engine with this identical input results in the same output. We considered that executing the runs with duplicated configurations and exclude the duplicated run results during the post-processing of the evaluation output would provide improved time-performance during the evaluation process. Altogether, the total number of experiments performed, regardless of the repetition of values, was 22,364. The number of suggestions ($|S_x|$), the elements of the S_x set, and their respective relevance indexes were stored. The *relevance index* was used to define a descending order for the categories in each S_x set. The value of $|R_x \cap S_x|$ was also stored for each one of the combinations.

The values were grouped by n, then by $|R_x|$, and finally by $|R_x \cap S_x|$ to calculate the descriptive statistics and perform analyses upon these values. A *Hit Rate* is given by the number of executions with the same configuration (e.g., $n = 2$ and $|R_x| = 5$) where $|R_x \cap S_x| \geq 1$ (thus, at least a partial success), divided by the total number of executions with that configuration. Table 4 shows the *Hit Rate* for each of 50 combinations varying n from 1 to 10, and $|R_x|$ from 1 to 5, and the values of the mean, median and standard deviation for:

- the number of suggested contextual element categories ($|S_x|$); and
- the average position of R_x's contextual element categories in S_x according to the *relevance index*, which is used to order S_x descendingly.

Regarding the *average position*, for example, given $R_x = [3, 5, 9]$ and $S_x = [9, 6, 3, 5, 7, 4]$, it is calculated as $(3 + 4 + 1)/3$, since 3 is positioned as the third element in S_x, 5 is the fourth element and 9 is the first one, so the result is 2.67. Observe that after ordering the results based on the relevance index, the sets are treated as lists, so the order of the elements is maintained.

Therefore, the **first objective** of the evaluation was to identify an **ideal value for n**. This value should allow the recommender system to select a sufficient number of projects similar to the *baseline project* to generate the list of recommendations for it. So, one expects that most of the categories removed from the baseline project are part of such a list.

Furthermore, the recommendation list should be short enough, with only the most probable useful categories. This way, engineers can analyze the recommended categories more carefully. Thus, the **second objective** was to answer the following question: **is it possible to limit the number of recommended categories without a negative impact on the quality of the recommendations?**

Table 4 – Hit rate and the mean, median, and standard deviation for the number of Recommended Contextual Elements Categories and Position of the Categories in the Recommendation List, by n and $|R_x|$

			No. of Recommended Contextual Elements Categories			Average Position of the Categories in the Recommendation List		
n	$\|R_x\|$	Hit Rate	Mean	Median	Std. Dev.	Mean	Median	Std. Dev.
1	1	25.3%	4.10	4.00	2.27	2.70	2.00	1.97
1	2	43.5%	4.28	4.00	2.13	2.85	2.00	1.78
1	3	53.3%	4.44	4.00	2.32	2.92	2.50	1.68
1	4	58.3%	4.81	5.00	2.33	3.10	3.00	1.82
1	5	61,5%	4,42	4,00	2,28	2,97	3,00	1,64
2	1	46,7%	7,21	7,00	2,98	3,89	3,00	2,96
2	2	67,1%	7,02	7,00	2,91	3,81	3,00	2,56
2	3	77,4%	7,98	8,00	3,08	4,41	4,00	2,46
2	4	81,7%	8,23	8,00	3,14	4,59	4,29	2,52
2	5	87,1%	8,46	8,00	3,16	4,68	4,33	2,34
3	1	53,2%	9,72	10,00	3,41	4,35	3,00	3,29
3	2	74,9%	10,21	10,00	3,60	4,78	4,00	3,21
3	3	85,0%	10,98	11,00	3,60	5,17	4,67	3,05
3	4	91,4%	11,26	11,50	3,45	5,33	5,00	2,95
3	5	91,6%	11,47	12,00	3,58	5,69	5,33	2,63
4	1	59,8%	12,25	12,00	3,99	4,70	4,00	3,48
4	2	83,1%	12,96	13,00	3,88	5,10	4,50	3,34
4	3	90,5%	13,86	14,00	4,15	5,99	5,33	3,59
4	4	95,0%	13,82	14,00	3,80	6,24	6,00	3,37
4	5	96,3%	14,74	15,00	3,93	6,93	6,50	3,36
5	1	69,3%	14,74	14,00	4,17	5,77	4,00	4,81
5	2	86,9%	15,28	15,00	4,18	5,67	5,00	4,09
5	3	93,1%	16,14	16,00	4,05	6,50	6,00	3,83
5	4	97,1%	16,61	17,00	4,15	7,26	7,00	3,57
5	5	97,5%	16,78	17,00	3,96	7,31	7,00	3,53
6	1	73,0%	16,62	16,00	4,24	5,66	4,00	4,73
6	2	91,1%	17,49	17,00	4,32	6,37	5,50	4,14
6	3	95,2%	18,05	18,00	4,34	7,13	6,50	4,22
6	4	97,4%	18,80	19,00	4,31	7,54	7,00	3,71
6	5	99,4%	19,20	19,50	4,16	8,10	7,75	3,60
7	1	76,6%	18,95	18,50	4,38	6,56	4,00	5,89
7	2	94,8%	19,41	19,00	4,22	7,04	6,00	4,82
7	3	98,6%	19,96	20,00	4,49	7,68	7,00	4,40
7	4	99,0%	20,56	21,00	4,32	8,04	7,67	3,93
7	5	99,7%	20,95	21,00	4,23	8,52	8,25	3,64
8	1	81,3%	20,67	20,00	4,69	6,78	5,00	5,94
8	2	94,5%	21,07	21,00	4,35	7,12	6,00	4,91
8	3	98,8%	21,73	22,00	4,41	7,78	7,33	4,17
8	4	99,2%	22,52	22,00	4,40	8,48	8,00	4,18
8	5	100,0%	23,10	23,00	4,21	9,00	8,55	3,64
9	1	83,8%	22,20	22,00	4,37	7,27	5,00	6,29
9	2	93,6%	22,86	23,00	4,25	7,42	6,50	4,70
9	3	98,3%	23,20	23,00	4,31	8,40	7,67	4,61
9	4	99,5%	24,01	24,00	4,26	8,72	8,50	3,97
9	5	100,0%	24,17	25,00	4,36	9,49	9,20	3,56
10	1	86,0%	23,83	24,00	4,40	7,37	5,00	6,57
10	2	96,0%	24,27	24,00	4,32	7,57	6,50	5,11
10	3	98,9%	25,18	25,00	4,61	8,94	8,33	4,68
10	4	99,5%	25,36	25,00	4,24	8,99	8,50	4,28
10	5	100,0%	25,85	26,00	4,38	9,71	9,33	3,81

Source: The author (2022)

4.4.1 The Ideal Value for n

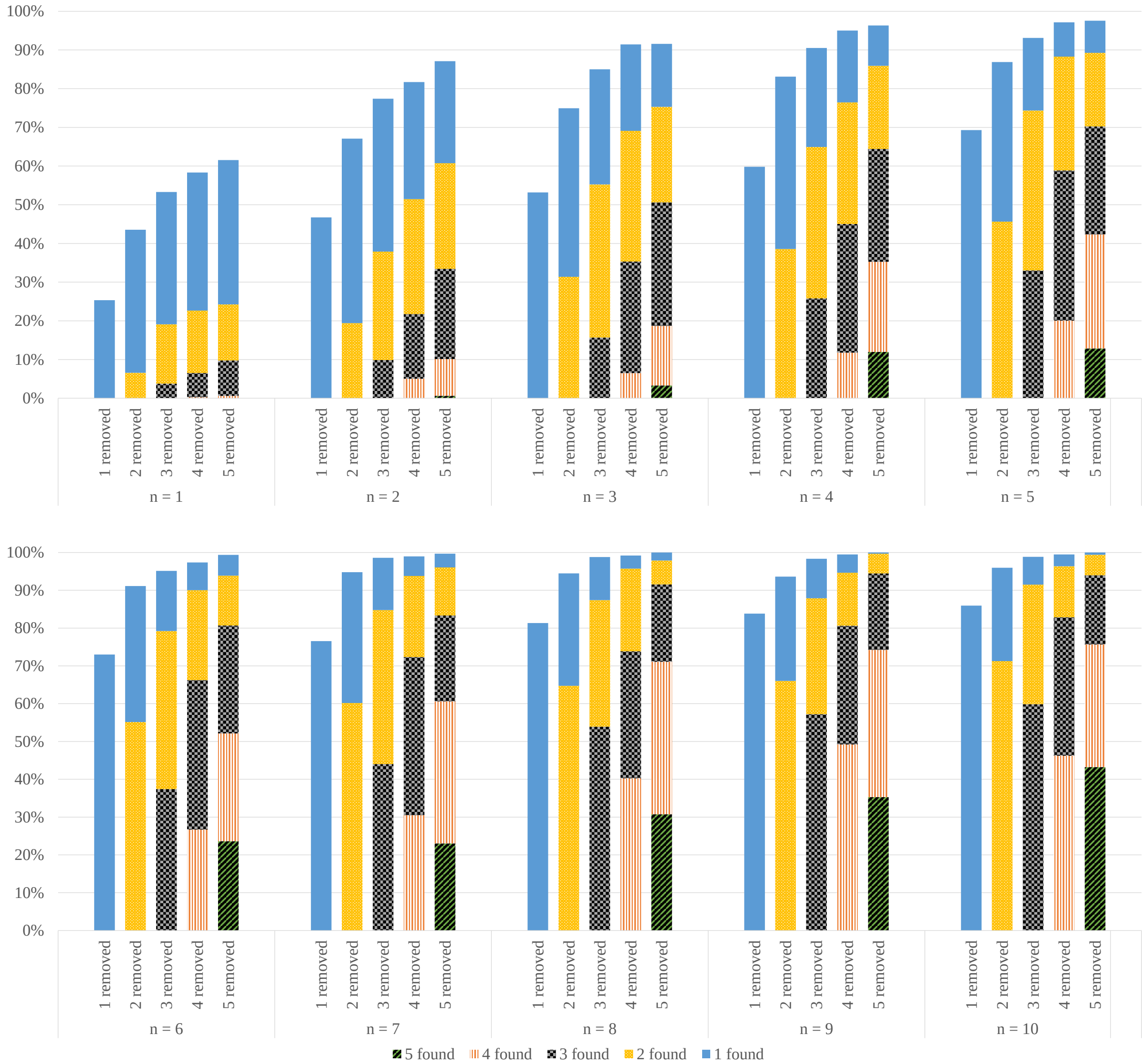
Given the objective of identifying whether there is an *ideal value for* n, the data were organized as a stacked bars chart seen in Figure 18. For each value of n, there is a group of 5 bars, one for each $|R_x|$. Each bar can have the number of sections up to $|R_x|$, and each section shows the number of categories in $R_x \cap S_x$. It is straightforward to observe that low n values are unable to provide a satisfactory result, with values as low as 25,3% of Hit Rate for $n = 1$ and $|R_x| = 1$. As the value of n increases, however, it is possible to observe that the Hit Rate also increases, surpassing 75% for $|R_x| = 1$ when $n \geq 7$. On the other hand, one expects the engine to have some difficulty suggesting all the removed contextual element categories when the amount of information removed is large (say $|R_x| \geq 3$). In this case, a higher value of $|R_x|$ implies the size reduction of the remaining set, so the engine can hardly identify similar projects. Nevertheless, when $n = 8$, the engine was able to suggest back all the removed categories in more than 50% of the runs for $|R_x| = 3$ and more than 30% for $|R_x| = 5$.

An analysis of the Hit Rate using this chart alone could indicate that higher values of n are preferred, and indeed at first sight a higher n generates better results in terms of Hit Rate, with higher Hit Rate values as can be seen in Table 4 and in Figure 18. However, since the results are supposed to be analyzed by a requirements engineers, another variable that must be considered to choose an ideal value of n is the total number of suggestions generated by the engine. If the recommender engine suggests a reasonably small number of categories to be analyzed, the developer will more easily investigate the fitness of elements from each suggested category. However, if this value is too high, this would translate into a high load of work to the responsible for analyzing whether the suggested categories indeed contain useful elements for the target application. Also, it is possible to conjecture that the high Hit Rate was achieved only by offering a large number of suggestions, with no regard to the quality of the suggestions themselves.

4.4.2 Defining a threshold relying on the relevance index

To verify whether a large value of n was creating an output with too many suggestions, another chart was plotted. Figure 19 (a) is a Box Plot representation of the number of suggestions made in all runs grouped by n. It is noticeable that as n increases, the number of

Figure 18 – Removed Contextual Element Categories (R_x) x Removed Contextual Element Categories in Suggestion List ($|R_x \cap S_x|$), by n

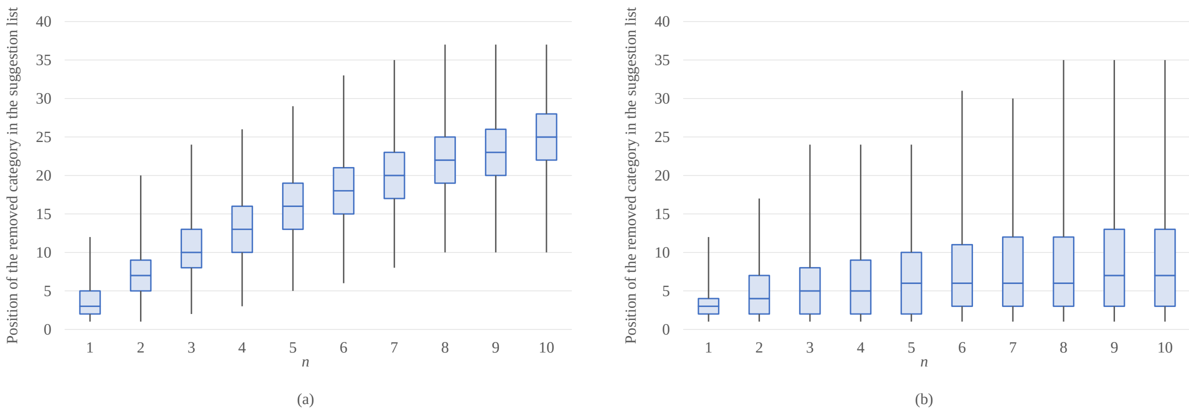


Source: The author (2022)

contextual element categories suggested also increases considerably. The values of n that had the best results when considering only the Hit Rate suggest five times more categories, with the runs with $n = 10$ suggesting a median of 25 categories, with at least one run for each $n \geq 8$ with 37 categories. These values would indicate an impracticability of using such high values of n, with an ideal n value less than 6. However, another factor that must be considered is the position where the suggestions which are part of $(Rx \cap S_x)$ appear in the S_x set ordered by their *relevance index*.

Figure 19 (b) illustrates the analysis that takes into account the position of the suggested categories according to the relevance index. It is also a Box Plot with the runs grouped by

Figure 19 – (a) Contextual Element Categories Suggested, by n, and (b) Position of Removed categories that were suggested, by n



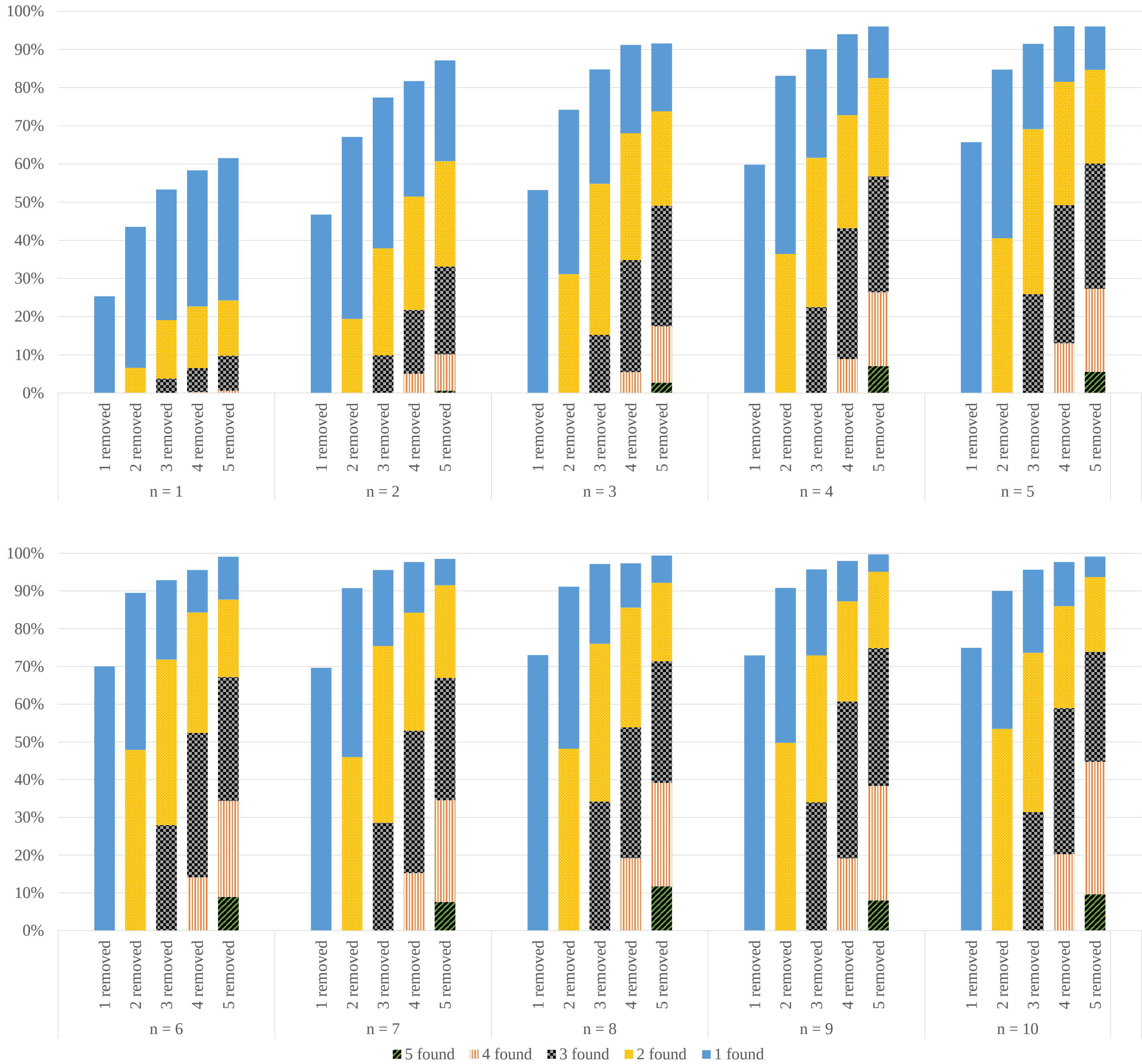
Source: The author (2022)

n, but this time we can observe that even for high values of n, 75% of the suggestions in $(R_x \cap S_x)$ appear before the 15th position, when ordered in descending order by the relevance index. This is a promising result, as it both validates that the *relevance index* is useful in giving priority to suggestions likely to be fit to the set of contextual element categories given as baseline, as it also points towards a *threshold* of categories that could trim the suggestion list. Using such a threshold could allow using higher values of n without the problem of offering a large number of suggestions to the engineer responsible for analyzing the viability of using elements from the suggested categories.

To measure the impact of using a threshold on the length of the suggestion list, all the runs previously executed had their suggestion lists trimmed to contain only their 15 first categories when ordered in descending order by the relevance index of each suggestion. Then, the Hit Rate was computed again, and Figure 20 was plotted based on the new data. One observes that there is indeed a negative impact on the Hit Rate, but it is small, not impairing the overall quality of the list returned by the recommender engine. The most significant impact is observed when $|R_x| \geq 3$, where the recommender engine using the threshold is much less frequently able to return all the removed categories. This happens since as $|R_x|$ increases, i.e., a higher number of categories are removed, the set of categories (E'_x) given as input to the recommender engine becomes less similar to the baseline project, making the *relevance index* less accurate. As a consequence, the order of the categories in the recommendation list is less reliable, and some of the recommended items that are elements of R_x will be positioned above the trimline (*threshold*) – e.g., 16$^{\text{th}}$ position and above.

The analysis of these results indicates that using high values of n (≥ 8) in conjunction

Figure 20 – Removed Contextual Element Categories (R_x) x Removed Contextual Element Categories in Suggestion List ($|R_x \cap S_x|$), by n, and limiting the suggestion list's length ($|S_x|$) to 15 (*threshold*)



Source: The author (2022)

with a threshold value (15) that caps the number of suggestions provided by the recommender engine according to the relevance index given in Equation 4.1 can provide interesting insights into valid contextual elements that are suitable for use together with the elements that compose the categories of the baseline set. This suggests that the tool would perform its intended function when the designers of new context-aware intelligent transportation systems use it informing an initial set of already known contextual elements that must be used in the application, to receive suggestions of other categories of useful contextual elements that were potentially missed during the analysis of the application requirements.

4.5 DISCUSSION

The data obtained from the validation experiment indicated that the designed recommender system achieves satisfactory results to suggest related contextual element categories when the number (n) of "most similar" projects selected is large enough, such as 8 or higher. This evaluation is based mostly on the *Hit Rate*, a commonly used evaluation metric in recommender systems according to Sridevi, Rao e Rao (2016). The Hit Rate is a valid and useful metric for recommender systems' accuracy.

The results also indicate that as the number of considered projects increases, the size of the suggestion set also increases, which can be a burden to the requirements engineers responsible for analyzing the recommended categories to identify contextual elements to be used in the ITS application being developed. This was an expected side-effect of increasing the number of projects used as most similar, and so we have designed the recommender system with a functionality to rank the suggestions in order of relevance. To accomplish this, we devised the *relevance index* described in Equation 4.1.

The analysis of the results confirms that the relevance index-based provides a good metric of how well the suggested contextual element category fits in the given input subset of categories. The high concentration of previously known-to-be-fit categories in the first positions of the suggestion lists recommended by our tool shows that. This approach for validating the relevance index-based ordering is a good fit for our research, given that the information about the expected recommendations is previously available. This avoids the need to use more complex ranking metrics that are difficult to communicate, while keeping the advantage of being able to test the order of the recommendations like the ranking metrics described by Sridevi, Rao e Rao (2016).

This last finding is important because it allows us to limit the number of contextual element categories returned by our tool, or at least providing information to the users that at least 75% of the categories are probably found up to the 15^{th} element of the list, for example. This *threshold* greatly reduces the already mentioned burden of analyzing lots of contextual element categories, giving more time for a deeper analysis on the most probably-useful categories.

Our final analysis on the quality of the suggestion lists after limiting their sizes to 15 elements shows that the negative impact on accuracy is minimal. The gains in time for analysis due to trimming more than 50% of the suggestion list can be considerable given the small reduction of accuracy. This led us to define the *threshold* for the recommendation list size to be 15 categories, so we can limit the output size without a noticeable quality loss.

4.6 FINAL CONSIDERATIONS

Overall, we expect the tool devised in this research can be useful to ITS application developers in the context of the software creation, allowing richer context to be used. Such a recommender system could be used either as a complement to the requirements elicitation process, or as a companion of developers in their first contact with the application requirements. It can also be used as a companion tool when following the guidelines proposed by Alegre-Ibarra, Augusto e Evans (2018), or also when using a method such as the one proposed by Engelenburg, Janssen e Klievink (2019).

The benefits of identifying early in the software development process the possibility of using more contextual elements might lead to improved ITS applications that are more likely to be adopted. In such a change-resistant industry, the automotive one, this early identification might be significant to include the contextual elements in the application design, once it is unlikely that substantial changes are introduced in many of the projects in this area after they reach more advanced stages of development.

5 CONCLUSION

The automotive industry more and more relies on software, and the number of ITS projects developed year after year is growing. Identifying the context-aware related requirements and the potential contextual elements that could benefit each of them is a non-trivial task. The knowledge extracted from the design decisions taken in those projects can help software professionals avoid missing important context information in the design of new systems.

Understanding which Contextual Elements are available and could be used is essential to application designers who intend to use context-awareness in their applications. Contextual models come to aid in this design process, providing insights into already known and used elements and categories, their semantics, and the relationships among them. This is directly connected to our main research question (RQ), stated in Section 1.2.1, since such a model can be a valid approach to improve the discovery of contextual elements when an application is being designed.

Our model intends to guide vehicular application designers and researchers to understand the possibilities that context can offer to their systems. By choosing a simple hierarchical notation, with the descriptions given in this work, we make the model accessible and useful to people without prior knowledge in context-aware techniques or in reading more complex, yet more powerful, representations such as ontologies.

Our categorization model uses the four primary context categories as the top-level categories, with the consideration that while there is no category for History-related elements, they are already represented in the other four and can be considered to be used both as live data and also as historical data, to fulfill the timeliness requirement that models must have. From these categories, we defined specialized subcategories to hold the types of contextual elements that our research has elicited as valid for vehicular applications, based on literature research of commonly used information in current context-aware vehicular applications. In our description, we suggest potential elements for each subcategory and mention the usual data gathering methods used for elements in each category. Validating the proposed taxonomy by designing a vehicular application was a good exercise to check for consistency and completeness of the model, which was useful to identify enhancement possibilities in the initial idea that was used to define the application.

The taxonomy was defined with the goal of being used in a recommender system that could automatically suggest potentially useful contextual element categories to system designers. This decision reduces the granularity of the items, so instead of Contextual Elements, Categories that represent several elements. This allows for more relations being identified in not so vast knowledge bases like ours.

Using a *recommender system* based on the knowledge of previously designed projects can be an alternative to improve the quality of new ITS designs. We designed and developed such a tool by using *cosine similarity* measurements between 57 ITS projects and conceived a *relevance index* based on the similarity value and the number of projects where a contextual element from a category is present. The process used in the recommender system completes the answer for our main research question (RQ).

To validate this tool and the relevance index, we devised a method that removed a project from the knowledge base and removed the information of some of the contextual element categories used in this project. We then fed it to the recommender system to check whether it would suggest back the removed contextual element categories. According to the results of our experiments (18,598 distinct runs), the proposed tool has proven to be valuable and provide useful results. To avoid an excessive list to be analyzed, it used the relevance index to limit the suggestion list to the 15 top-ranked categories. Our evaluation showed that this threshold did not have a meaningful negative impact on the resulting recommendation. Using $n \geq 7$, the recommender system was still able to recommend more than 50% of the removed contextual element categories in more than 75% of the runs, reaching more than 92% of the runs in the case of $n = 10$ and $|R_x| = 5$.

Thus, we confirmed that it is possible, given an incomplete set of categories know to be needed by an application, to recommend specific Contextual Element Categories with success. Also, we could verify that we can limit the number of recommended categories to allow an efficient analysis by software development professionals with a low impact on the quality and completeness of this recommendation.

The development of the knowledge-base and the taxonomy guided us on answering our secondary research question RQ1. The contextual elements identified are listed in Appendix C, and are classified in our proposed taxonomy. An answer to our secondary question RQ2 is the automation level of the recommender. After the initial user input of known contextual elements needed for an application, the recommender automatically uses the information from the knowledge base to provide ordered recommendations to that instance of the problem.

5.1 CONTRIBUTIONS

The research reported in this thesis provides contributions to the design of context-aware ITS applications and the organization of knowledge on how context is used in the ITS domain. We can group the major contributions of this work in the three topics that will be presented in the following.

5.1.1 A Taxonomy for ITS Contextual Elements

The process that led to the definition of a taxonomy for ITS Contextual Elements was presented in Chapter 3. This taxonomy defines 79 categories in total, with the 4 top-level categories (called *supra-categories* in this work) being Activity, Identity, Location and Time. To the best of our knowledge no other similar work in the same domain and level of granularity exists to map usage of Contextual Elements in ITS projects.

The taxonomy itself is valuable in representing in a didactic way how ITS applications can benefit from contextual elements. Its level of granularity, representing contextual categories rather than elements, reduces the information overload when a person is in charge of manually analyzing the possibilities of their applications, or just wants to understand how context is used in applications in this domain.

Used in conjunction with the other artifacts and contributions generated during this research, such as the list of contextual elements in each category (available in Appendix C), or the knowledge base and recommender system that will be discussed next, this taxonomy has the potential to contribute to the design of improved ITS applications.

5.1.2 A knowledge base of context-awareness usage in ITS applications

The knowledge base consisting of ITS projects and their usage of contextual element categories is another major contribution of this work. The collection of 70 academic projects and additional 3 commercial applications, with a mapping of which contextual element categories are used in each one of them can have uses in industry or academia. Software professionals might be able to quickly find projects that use elements of a specific type. In the academic environment, future research can use the information contained in the knowledge base to investigate further into the ITS domain.

This knowledge base is also not a static entity, and can be improved and augmented by future researchers with more information from different new projects. As long as there is a careful analysis on the usage of contextual elements in each new project to be added, it is expected that the knowledge base keeps being helpful to the goal of providing insightful information of how ITS projects handle context-awareness in terms of the contextual elements used.

5.1.3 A process to provide automatic recommendations of Contextual Element Categories to ITS

The most relevant contribution of this work is the process designed to recommend contextual element categories to ITS during their design. The details about the design and evaluation of the recommender system of contextual element categories for ITS applications proposed in this work are presented in Chapter 4.

Overall, the recommender system operates based on the similarity between a *base project* - a set of contextual element categories that are already known to be needed in the ITS project being designed - and the projects that are part of the knowledge base described in the previous subsection. This similarity is based on which contextual element categories are used in each project, and is calculated using cosine similarity. After identifying the most similar projects, the engine looks for the contextual element categories used in them that are not part of the base project. It ranks them using our proposed relevance index, which is a utility function to rank the recommendations in order of how likely they are to be useful to that particular project.

Our evaluation process used the information of the knowledge base to assess whether the proposed recommender system would provide useful results. It worked by taking out a project from the knowledge base, removing some of the contextual element categories from its list, and using this new set of contextual element categories as the base project to the recommender system. When at least the 8 more similar projects are used in the stage of finding the most similar projects, we can limit the number of recommendations to 15 and the system still suggests the categories that were removed from the baseline project in 75% of the times. This indicates that most of the time the system is able to recommend categories that were used in the actual project.

This recommender system can be used by engineers when designing their applications after some initial requirements elicitation have happened. In such use, the contextual elements found during the requirements phase can be mapped to their respective categories in our taxonomy, and then inputted as a base project in the recommender system to find, using the underlying knowledge base, other potentially useful categories for such a project. The knowledge base can be changed to include proprietary projects of the user, or further expanded as mentioned in the previous subsection, and the overall process of the recommender system will still work the same way.

5.2 LIMITATIONS

While we have promising results that were discussed earlier, throughout this research some potential improvements or limitations were identified. In the following list we present these topics:

- The knowledge base can be incremented to include more projects. Finding relevant ITS projects and identifying the categories used in them is not trivial and very time-consuming. As mentioned before, many ITS projects take context-awareness for grant, and do not mention use of contextual elements, or even context, in their respective manuscripts. While the results obtained from our validation of the recommender system pointed that its suggestions are useful, we expect that a larger knowledge base could provide even better results.
- Most of the projects do not use the same notations and definitions of context-awareness as the one adopted in this research. Mapping the contextual elements based on what is reported in the articles that present the projects is a manual process. Even though that was carefully conducted, it is error-prone, mostly in the case of missing that a contextual element is used in a project. However, the error of misidentifying a contextual element that is not used in a project is expected to be rare, and is not currently considered a threat.
- We were not able to validate the results of the proposal with real experienced engineers in the ITS domain. Our statistical and empirical analysis of the results suggests that our approach is promising in the scenario we described. However, we recognize that this assumption can only be confirmed if tested with a significant number of engineers involved in the design of applications of the ITS domain.

5.3 FUTURE WORKS

The results achieved in this research allow for several possible lines of evolution. In both of the major contributions, the taxonomy and the recommender system, there are numerous potential future works to further validate possible additions or improvements to the artifacts that were designed and built during this research.

Regarding the taxonomy, we consider that a promising future work is the creation of an ontology based on our hierarchical model. While the hierarchical model has advantages that come from its simplicity, as mentioned in Chapter 3, it lacks the power for use by computers directly. The expressiveness of an ontology, with its unbounded relations, could be useful to represent richer structures. An evolution of the taxonomy to an ontology could even contain the very contextual elements represented in the model, and not only the categories.

Another potential work is the analysis of how to include Quality of Information (QoI) features in the taxonomy, or in its ontology evolution. QoI, in a broad sense, is "the extent to which the data corresponds to the real world" (EBLING; HUNT; LEI, 2001), or in a more strict definition, is "the body of tangible evidence available that can be used to make judgments about the fitness-of-use and utility of information products" (BISDIKIAN et al., 2009). Representing QoI requirements and constraints in the model, especially if using an ontology, could provide more insightful information to engineers using it. More research however is required to validate whether it could really be useful, and to explore the potential of such an addition.

Another potential evolution of the categorization model for contextual elements is in the direction of its generalization. Either in its current form of a taxonomy, or after its evolution to an ontology, there is the possibility for research on the definition of a more general structure, encompassing domains other than ITS. This could allow the creation of a generic upper-level contextual model, which could be used to share information and communicate context-aware application from different domains.

Regarding the recommender system, it would be useful to perform a new execution of the validation experiments, this time using the most updated versions of the taxonomy and of the knowledge base, which is described in Section 3.5.4. Other than the newer version of the taxonomy and the knowledge base, the protocol for this execution must be the same one used in this research, so we can compare the results. We expect that the same overall results hold, or even that with more projects, either the n number of most-similar projects to achieve the 75% rate of success can be reduced from 8.

Also, an interesting work could be to implement other similarity measurements in the recommender system and compare their results with the results we obtained using cosine similarity. Although some research indicates that the differences among the candidate similarity measurement techniques are not so relevant for some recommender systems (LATHIA; HAILES; CAPRA, 2008), other research delve deep into the comparison of similarity measurement techniques, comparing their results and overall performance (FKIH, 2021), so an investigation using our recommender system as a testbed could provide insightful results.

Another future work involves considering the published year of a project in the calculation of the relevance index. This could be used to favor newer projects, which could be more adherent to current practices and tendencies. An evaluation comparing the results of this recency-biased relevance index to the current results is also an essential part of this proposed future work.

Still on the recommender system, an update of the GUI and the underlying knowledge base with the categories and projects in the version 1.1 of the taxonomy is also a future work. Changing the current infrastructure of how the knowledge base that is used in the system is stored could be useful to allow for easier adding and removal of items later. This would allow the creation of a tool where the user can edit the knowledge base. The user could both add new projects to the knowledge base, as well as disable existing projects from being used. Currently, changing the knowledge base involves editing the template R script to add new projects or remove existing ones, which can be error-prone, since multiple points of the file require changes. Such an improvement could be useful for companies willing to use the recommender system taking advantage of private information regarding their internal projects, for instance.

Still on the GUI of the recommender system, providing real-world examples of usages of contextual elements of the suggested categories could be a powerful addition to the system. It would ease the analysis of the requirements engineers who are using the recommender system, as well as guide them on proper choices based on past experiences from other projects in the domain.

Another interesting future research project involves further validation of the recommender system in a real environment of software development. A case study involving the whole process of developing an ITS application, similar to the scenario presented in Section 1.1, but using the contributions proposed in this work, such as the recommender system of contextual element categories and its underlying knowledge base, would be able to validate whether our expectations of usage are strong enough. It could also check whether unexpected usages of the tools would arise, which on their turn could become new future researches.

Our secondary research question RQ3, *is it viable to generalize the approach to other domains*, is to be answered as a future work. A more ambitious and time-consuming future research could be carried out to validate this question. It would follow the overall methodology used in this work to build a taxonomy and a knowledge base to the contextual elements of a domain other than ITS, to validate whether our results could be generalized to other domains of context-aware applications.

REFERENCES

ABOWD, G. D.; MYNATT, E. D. Charting past, present, and future research in ubiquitous computing. *ACM Transactions on Computer-Human Interaction*, Association for Computing Machinery (ACM), v. 7, n. 1, p. 29–58, 3 2000.

ADOMAVICIUS, G.; TUZHILIN, A. Context-Aware Recommender Systems. *Recommender Systems Handbook*, Springer, p. 191–226, 2015. ISSN 978-1-4899-7636-9.

AGGARWAL, C. C. *Recommender systems*. [S.l.]: Springer, 2016. v. 1.

AGUIRRE, E.; LOPEZ-ITURRI, P.; AZPILICUETA, L.; REDONDO, A.; ASTRAIN, J. J.; VILLADANGOS, J.; BAHILLO, A.; PERALLOS, A.; FALCONE, F. Design and Implementation of Context Aware Applications with Wireless Sensor Network Support in Urban Train Transportation Environments. *IEEE Sensors Journal*, Institute of Electrical and Electronics Engineers Inc., v. 17, n. 1, p. 169–178, 2017. ISSN 1530437X.

AL-SULTAN, S.; AL-BAYATTI, A. H.; ZEDAN, H. Context-aware driver behavior detection system in intelligent transportation systems. *IEEE Transactions on Vehicular Technology*, Institute of Electrical and Electronics Engineers Inc., v. 62, n. 9, p. 4264–4275, 2013. ISSN 00189545.

ALAZAWI, Z.; ALTOWAIJRI, S.; MEHMOOD, R.; ABDLJABAR, M. B. Intelligent disaster management system based on cloud-enabled vehicular networks. In: *2011 11th International Conference on ITS Telecommunications, ITST 2011*. [S.l.: s.n.], 2011. p. 361–368. ISBN 9781612846712.

ALDAYEL, M.; BENHIDOUR, H. Product Recommendation in Case-based Reasoning. *2nd International Conference on Computer Applications and Information Security, ICCAIS 2019*, Institute of Electrical and Electronics Engineers Inc., 5 2019.

ALEGRE-IBARRA, U.; AUGUSTO, J. C.; EVANS, C. Perspectives on engineering more usable context-aware systems. *Journal of Ambient Intelligence and Humanized Computing*, Springer Verlag, v. 9, n. 5, p. 1593–1609, 10 2018. ISSN 18685145. Disponível em: <https://link.springer.com/article/10.1007/s12652-018-0863-7>.

ALEGRE, U.; AUGUSTO, J. C.; CLARK, T. Engineering context-aware systems and applications: A survey. *Journal of Systems and Software*, Elsevier, v. 117, p. 55–83, 2016. ISSN 0164-1212.

ALGHAMDI, W. Improving driver's behavior using Context-Aware Systems. *Procedia Computer Science*, Elsevier B.V., v. 10, p. 1213–1216, 1 2012. ISSN 18770509.

ALGHAMDI, W.; SHAKSHUKI, E.; SHELTAMI, T. R. Context-aware driver assistance system. *Procedia Computer Science*, Elsevier B.V., v. 10, p. 785–794, 1 2012. ISSN 18770509.

ALHAMMAD, A.; SIEWE, F.; AL-BAYATTI, A. H. An InfoStation-based context-aware on-street parking system. In: *2012 International Conference on Computer Systems and Industrial Informatics, ICCSII 2012*. Sharjah, United Arab Emirates: Institute of Electrical and Electronics Engineers, 2012. ISBN 9781467351577.

ALI, H.; MUHAMMAD, A.; KHAN, M. M. A Simple Framework for Context-Aware Driver Performance. *2020 IEEE 23rd International Conference on Intelligent Transportation Systems, ITSC 2020*, Institute of Electrical and Electronics Engineers Inc., 9 2020.

AMATRIAIN, X.; JAIMES, A.; OLIVER, N.; PUJOL, J. M. Data Mining Methods for Recommender Systems. In: *Recommender Systems Handbook*. Springer US, 2011. p. 39–71. Disponível em: <https://link.springer.com/chapter/10.1007/978-0-387-85820-3_2>.

ARKIAN, H. R.; ATANI, R. E.; KAMALI, S. Cluster-based traffic information generalization in vehicular ad-hoc networks. In: *2014 7th International Symposium on Telecommunications (IST 2014)*. Tehran, Iran: Institute of Electrical and Electronics Engineers, 2014. p. 1195–1200. ISBN 9781479953592.

ARNAOUTAKI, K.; MAGOUTAS, B.; BOTHOS, E.; MENTZAS, G. A Hybrid Knowledge-based Recommender for Mobility-as-a-Service. In: *ICETE (1)*. [S.l.: s.n.], 2019. p. 101–109.

ATASOY, B.; IKEDA, T.; SONG, X.; BEN-AKIVA, M. E. The concept and impact analysis of a flexible mobility on demand system. *Transportation Research Part C: Emerging Technologies*, Elsevier, v. 56, p. 373–392, 2015. ISSN 0968-090X.

BAE, I. H.; OLARIU, S. A tolerant context-aware driver assistance system for VANETs-based smart cars. In: *2010 IEEE Global Telecommunications Conference (GLOBECOM 2010)*. Miami, FL, USA: [s.n.], 2010. ISBN 9781424456383.

BAHRAMIAN, Z.; ABBASPOUR, R. A.; CLARAMUNT, C. A Cold Start Context-Aware Recommender System for Tour Planning Using Artificial Neural Network and Case Based Reasoning. *Mobile Information Systems*, Hindawi Limited, v. 2017, 2017. ISSN 1875905X.

BALTRUNAS, L.; KAMINSKAS, M.; LUDWIG, B.; MOLING, O.; RICCI, F.; AYDIN, A.; LÜKE, K.-H.; SCHWAIGER, R. Incarmusic: Context-aware music recommendations in a car. In: *International Conference on Electronic Commerce and Web Technologies*. Toulouse, France: Springer, 2011. p. 89–100.

BARAS, S.; SAEED, I.; TABAZA, H. A.; ELHADEF, M. VANETs-based intelligent transportation systems: An overview. In: *Lecture Notes in Electrical Engineering*. Springer Verlag, 2018. v. 474, p. 265–273. ISBN 9789811076046. Disponível em: <https://link.springer.com/chapter/10.1007/978-981-10-7605-3_44>.

BARBA, C. T.; AGUIAR, L. U.; IGARTUA, M. A.; PARRA-ARNAU, J.; REBOLLO-MONEDERO, D.; FORNÉ, J.; PALLARÈS, E. A collaborative protocol for anonymous reporting in vehicular ad hoc networks. *Computer Standards and Interfaces*, v. 36, n. 1, p. 188–197, 11 2013. ISSN 09205489. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S0920548913000615>.

BAUER, C.; NOVOTNY, A. A consolidated view of context for intelligent systems. *Journal of Ambient Intelligence and Smart Environments*, IOS Press, v. 9, n. 4, p. 377–393, 2017. ISSN 18761364.

BERGAN, A. T.; BUSHMAN, R. J.; TAYLOR, B. Intelligent truck rollover advisory systems. In: VRIES, M. J. de; KACHROO, P.; OZBAY, K.; CHACHICH, A. C. (Ed.). *Intelligent Transportation Systems*. Pittsburgh, PA, USA: SPIE, 1998. v. 3207, p. 140–147. Disponível em: <http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=932081>.

BETTINI, C.; BRDICZKA, O.; HENRICKSEN, K.; INDULSKA, J.; NICKLAS, D.; RANGANATHAN, A.; RIBONI, D. A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing*, Elsevier, v. 6, n. 2, p. 161–180, 2010.

BIFULCO, F.; AMITRANO, C. C.; TREGUA, M. Driving smartization through intelligent transport. *Chinese Business Review*, David Publishing Company, Inc., v. 13, n. 4, 2014. ISSN 1537-1506.

BISDIKIAN, C.; KAPLAN, L. M.; SRIVASTAVA, M. B.; THORNLEY, D. J.; VERMA, D.; YOUNG, R. I. Building principles for a quality of information specification for sensor information. In: *2009 12th International Conference on Information Fusion*. [S.l.: s.n.], 2009. p. 1370–1377.

BOBADILLA, J.; ORTEGA, F.; HERNANDO, A.; BERNAL, J. A collaborative filtering approach to mitigate the new user cold start problem. *Knowledge-Based Systems*, Elsevier, v. 26, p. 225–238, 2 2012. ISSN 0950-7051.

BOEHM, B. A view of 20th and 21st century software engineering. In: *Proceedings of the 28th International Conference on Software Engineering*. Shanghai, China: IEEE Computer Society, 2006. v. 2006, p. 12–29. ISBN 978-1-59593-375-1.

CAPELLEVEEN, G. van; AMRIT, C.; YAZAN, D. M.; ZIJM, H. The recommender canvas: A model for developing and documenting recommender system design. *Expert Systems with Applications*, Pergamon, v. 129, p. 97–117, 9 2019. ISSN 0957-4174.

CHAGAS, A. B.; FERRAZ, C. A. G. Smart Vehicles for Smarter Cities: Context-Aware V2X to Improve QoI. In: *Anais do XXIII Simpósio Brasileiro de Sistemas Multimídia e Web: Workshops e Pôsteres - WebMedia '17*. Gramado, Brasil: [s.n.], 2017. p. 64–69.

CHAVHAN, P. G.; CHAVHAN, G. H.; SHAFI, P. M.; MAHALLE, P. N. Context Awareness Computing System—A Survey. *Lecture Notes in Networks and Systems*, Springer, Singapore, v. 192 LNNS, p. 423–429, 2021. ISSN 23673389. Disponível em: <https://link.springer.com/chapter/10.1007/978-981-33-6546-9_40>.

CHAVHAN, S.; GUPTA, D.; CHANDANA, B. N.; KHANNA, A.; RODRIGUES, J. J. IoT-Based Context-Aware Intelligent Public Transport System in a Metropolitan Area. *IEEE Internet of Things Journal*, Institute of Electrical and Electronics Engineers Inc., v. 7, n. 7, p. 6023–6034, 7 2020.

CHAVHAN, S.; GUPTA, D.; NAGARAJU, C.; A, R.; KHANNA, A.; RODRIGUES, J. J. An Efficient Context-Aware Vehicle Incidents Route Service Management for Intelligent Transport System. *IEEE Systems Journal*, Institute of Electrical and Electronics Engineers Inc., 2021.

CHEN, T.; LU, S. Context-aware lane marking detection on urban roads. In: *Proceedings - International Conference on Image Processing, ICIP*. [S.l.]: IEEE Computer Society, 2015. v. 2015-December, p. 2557–2561. ISBN 9781479983391. ISSN 15224880.

DAVID, B.; XU, T.; JIN, H.; ZHOU, Y.; CHALON, R.; ZHANG, B.; YIN, C.; WANG, C. User-oriented system for smart city approaches. In: *12th IFAC Symposium on Analysis, Design, and Evaluation of Human-Machine Systems*. [S.l.]: Elsevier, 2013. v. 46, n. 15, p. 333–340. ISBN 9783902823410. ISSN 14746670.

DEY, A.; ABOWD, G.; SALBER, D. A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. *Human-Computer Interaction*, L. Erlbaum Associates Inc., v. 16, n. 2, p. 97–166, 12 2001.

DEY, A. K. Understanding and Using Context. *Personal and Ubiquitous Computing*, Springer Nature, v. 5, n. 1, p. 4–7, 2 2001.

DIBAEI, M.; ZHENG, X.; JIANG, K.; ABBAS, R.; LIU, S.; ZHANG, Y.; XIANG, Y.; YU, S. Attacks and defences on intelligent connected vehicles: a survey. *Digital Communications and Networks*, Elsevier, v. 6, n. 4, p. 399–421, 11 2020. ISSN 2352-8648.

DIKAIAKOS, M. D.; FLORIDES, A.; NADEEM, T.; IFTODE, L. Location-Aware Services over Vehicular Ad-Hoc Networks Using Car-to-Car Communication. *IEEE Journal on Selected Areas in Communications*, IEEE Press, Piscataway, NJ, USA, v. 25, n. 8, p. 1590–1602, 10 2007.

DUMITRU, H.; GIBIEC, M.; HARIRI, N.; CLELAND-HUANG, J.; MOBASHER, B.; CASTRO-HERRERA, C.; MIRAKHORLI, M. On-demand Feature Recommendations Derived from Mining Public Product Descriptions. In: *Proceeding of the 33rd international conference on Software engineering - ICSE '11*. New York, New York, USA: ACM Press, 2011. p. 181–190. ISBN 9781450304450.

DZEMYDIENĖ, D.; BURINSKIENĖ, A. Integration of Context Awareness in Smart Service Provision System Based on Wireless Sensor Networks for Sustainable Cargo Transportation. *Sensors 2021, Vol. 21, Page 5140*, Multidisciplinary Digital Publishing Institute, v. 21, n. 15, p. 5140, 7 2021. Disponível em: <https://www.mdpi.com/1424-8220/21/15/5140/htmhttps://www.mdpi.com/1424-8220/21/15/5140>.

EBLING, M. R.; HUNT, G. D. H.; LEI, H. Issues for context services for pervasive computing. In: *Proc. Advanced Topic Workshop Middleware for Mobile Computing, IFIP/ACM Middleware 2001 Conference*. [s.n.], 2001. p. 55–65. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.417.1618>.

ELHADEF, M. An adaptable inVANETs-based intersection traffic control algorithm. In: *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*. Liverpool, UK: [s.n.], 2015. p. 2387–2392.

ENGELENBURG, S. van; JANSSEN, M.; KLIEVINK, B. Designing context-aware systems: A method for understanding and analysing context in practice. *Journal of Logical and Algebraic Methods in Programming*, Elsevier, v. 103, p. 79–104, 2019. ISSN 2352-2208.

ENGIN, G.; AKSOYER, B.; AVDAGIC, M.; BOZANLI, D.; HANAY, U.; MADEN, D.; ERTEK, G. Rule-based expert systems for supporting university students. *Procedia Computer Science*, Elsevier B.V., v. 31, p. 22–31, 1 2014. ISSN 18770509.

Environmental Protection Agency. *Code of Federal Regulations. 40 CFR 156.62 - Toxicity Category*. 2021. Disponível em: <https://www.ecfr.gov/current/title-40/chapter-I/subchapter-E/part-156/subpart-D/section-156.62>.

ESSIEN, A.; PETROUNIAS, I.; SAMPAIO, P.; SAMPAIO, S. A deep-learning model for urban traffic flow prediction with traffic events mined from twitter. *World Wide Web*

2020 24:4, Springer, v. 24, n. 4, p. 1345–1368, 3 2020. ISSN 1573-1413. Disponível em: <https://link.springer.com/article/10.1007/s11280-020-00800-3>.

FALCÃO, R.; TRAPP, M.; VIEIRA, V.; SILVA, A. Vianna Dias da. Using a Data-Driven Context Model to Support the Elicitation of Context-Aware Functionalities – A Controlled Experiment. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer, Cham, v. 13126 LNCS, p. 119–135, 2021. ISSN 16113349. Disponível em: <https://link.springer.com/chapter/10.1007/978-3-030-91452-3_8>.

FALCAO, R.; VILLELA, K.; VIEIRA, V.; TRAPP, M.; FARIA, I. L. D. The practical role of context modeling in the elicitation of context-aware functionalities: A survey. *Proceedings of the IEEE International Conference on Requirements Engineering*, IEEE Computer Society, p. 35–45, 2021. ISSN 23326441.

FALK, K. *Practical recommender systems*. [S.l.]: Simon and Schuster, 2019. ISBN 1638353980.

FARIA, R.; BRITO, L.; BARAS, K.; SILVA, J. Smart mobility: A survey. In: *Internet of Things for the Global Community, IoTGC 2017 - Proceedings*. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2017. p. 1–8. ISBN 9781538620649.

Federal Highway Administration. *Traffic Monitoring Guide - FHWA-PL-17-003*. [S.l.], 2016. Disponível em: <https://www.fhwa.dot.gov/policyinformation/tmguide/#app4c>.

FELFERNIG, A.; FRIEDRICH, G.; JANNACH, D.; ZANKER, M. Developing Constraint-based Recommenders. *Recommender Systems Handbook*, Springer, Boston, MA, p. 187–215, 2011. Disponível em: <https://link.springer.com/chapter/10.1007/978-0-387-85820-3_6>.

FELFERNIG, A.; ISAK, K.; SZABO, K.; ZACHAR, P. The VITA financial services sales support environment. In: *Proceedings of the national conference on artificial intelligence*. [S.l.]: Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2007. v. 22, n. 2, p. 1692.

FELFERNIG, A.; TEPPAN, E.; GULA, B. Knowledge-based recommender technologies for marketing and sales. *International Journal of Pattern Recognition and Artificial Intelligence*, World Scientific Publishing Company, v. 21, n. 2, p. 333–354, 3 2007. ISSN 02180014.

FIGUEIREDO, L.; JESUS, I.; MACHADO, J. A. T.; FERREIRA, J. R.; CARVALHO, J. L. M. de. Towards the development of intelligent transportation systems. In: *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No.01TH8585)*. Oakland, CA, USA: IEEE, 2001. p. 1206–1211.

FKIH, F. Similarity measures for Collaborative Filtering-based Recommender Systems: Review and experimental comparison. *Journal of King Saud University - Computer and Information Sciences*, Elsevier, 9 2021. ISSN 1319-1578.

FOGUE, M.; GARRIDO, P.; MARTINEZ, F. J.; CANO, J. C.; CALAFATE, C. T.; MANZONI, P.; SANCHEZ, M. Prototyping an automatic notification scheme for traffic accidents in vehicular networks. In: *4th IFIP Wireless Days*. Niagara Falls, ON, Canada: [s.n.], 2011. v. 1. ISBN 9781457720277. ISSN 21569711.

FUCHS, S.; RASS, S.; KYAMAKYA, K. A constraint-based and context-aware overtaking assistant with fuzzy-probabilistic risk classification. In: *IADIS International Conference Wireless Applications and Computing*. Amsterdam, The Netherlands: [s.n.], 2008. p. 93–100.

GEMMIS, M. D.; LOPS, P.; MUSTO, C.; NARDUCCI, F.; SEMERARO, G. Semantics-aware content-based recommender systems. In: *Recommender Systems Handbook, Second Edition*. Springer US, 2015. p. 119–159. ISBN 9781489976376. Disponível em: <https://link.springer.com/chapter/10.1007/978-1-4899-7637-6_4>.

GENA, C.; TORRE, I. The importance of adaptivity to provide onboard services: A preliminary evaluation of an adaptive tourist information service onboard vehicles. *Applied Artificial Intelligence*, Taylor & Francis Group, v. 18, n. 6, p. 549–580, 7 2004. ISSN 08839514. Disponível em: <https://www.tandfonline.com/doi/abs/10.1080/08839510490463442>.

GHAFFARIAN, H.; FATHY, M.; SORYANI, M. Vehicular ad hoc networks enabled traffic controller for removing traffic lights in isolated intersections based on integer linear programming. *IET Intelligent Transport Systems*, v. 6, n. 2, p. 115–123, 6 2012. ISSN 1751956X.

GOLESTAN, K.; SOUA, R.; KARRAY, F.; KAMEL, M. S. Situation awareness within the context of connected cars: A comprehensive review and recent trends. *Information Fusion*, Elsevier, v. 29, p. 68–83, 5 2016.

GOMES, J. Z.; BARBOSA, J. L. V.; GEYER, C. F. R.; ANJOS, J. C. S. D.; CANTO, J. V.; PESSIN, G. Ubiquitous intelligent services for vehicular users: A systematic mapping. *Interacting with Computers*, Oxford University Press, v. 31, n. 5, p. 465–479, 2 2020. ISSN 09535438. Disponível em: <https://academic.oup.com/iwc/article/31/5/465/5618738>.

GOTO, K.; KAMBAYASHI, Y. Study on mobile passenger support systems for public transportation using multi-channel data dissemination. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer Verlag, 2002. v. 2362, p. 317–330. ISBN 3540439633. ISSN 16113349. Disponível em: <https://link.springer.com/chapter/10.1007/3-540-45636-8_25>.

GUERRERO-IBANEZ, J. A.; ZEADALLY, S.; CONTRERAS-CASTILLO, J. Integration challenges of intelligent transportation systems with connected vehicle, cloud computing, and internet of things technologies. *IEEE Wireless Communications*, Institute of Electrical and Electronics Engineers (IEEE), v. 22, n. 6, p. 122–128, 12 2015.

HAN, J.; KAMBER, M.; PEI, J. Getting to Know Your Data. In: *Data Mining*. [S.l.]: Elsevier, 2012. p. 39–82.

HAQUE, H. M. U.; ZULFIQAR, H.; KHAN, S. U.; HAQUE, M. U. Context-Aware Parking Systems in Urban Areas: A Survey and Early Experiments. *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, Springer, Cham, v. 266, p. 25–35, 11 2018. Disponível em: <https://link.springer.com/chapter/10.1007/978-3-030-06152-4_3>.

HAUPT, N. B.; LIGGESMEYER, P. Towards Context-Awareness for Enhanced Safety of Autonomous Vehicles. *Lecture Notes in Networks and Systems*, Springer, Cham, v. 295, p. 549–563, 9 2021. ISSN 23673389. Disponível em: <https://link.springer.com/chapter/10.1007/978-3-030-82196-8_40>.

HENRICKSEN, K.; INDULSKA, J. Developing context-aware pervasive computing applications: Models and approach. *Pervasive and mobile computing*, Elsevier, v. 2, n. 1, p. 37–64, 2006. ISSN 1574-1192.

HONG, J. y.; SUH, E. h.; KIM, S. J. Context-aware systems: A literature review and classification. *Expert Systems with Applications*, Pergamon, v. 36, n. 4, p. 8509–8522, 5 2009. ISSN 09574174.

HOOGENDOORN, R. G.; BREUKINK, H. J.; AREM, B. V. A context aware intelligent speed adaptation system: A field operational test. In: *IEEE Conference on Intelligent Transportation Systems, Proceedings (ITSC 2012)*. Anchorage, AK, USA: [s.n.], 2012. p. 1091–1096. ISBN 9781467330640.

HU, X.; LI, X.; NGAI, E. C.; LEUNG, V. C.; KRUCHTEN, P. Multidimensional context-aware social network architecture for mobile crowdsensing. *IEEE Communications Magazine*, Institute of Electrical and Electronics Engineers Inc., v. 52, n. 6, p. 78–87, 2014.

HU, X.; LI, X.; NGAI, E. C.; ZHAO, J.; LEUNG, V. C.; NASIOPOULOS, P. Health drive: Mobile healthcare onboard vehicles to promote safe driving. In: *Proceedings of the Annual Hawaii International Conference on System Sciences*. [S.l.]: IEEE Computer Society, 2015. v. 2015-March, p. 3074–3083. ISBN 9781479973675. ISSN 15301605.

HU, Y. L.; WANG, C. Y.; KAO, C. K.; CHANG, S. Y.; WEI, D. S.; HUANG, Y.; CHEN, I. Y.; KUO, S. Y. Toward fog-based event-driven services for internet of vehicles: Design and evaluation. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer Verlag, 2017. v. 10689 LNCS, p. 201–212. ISBN 9783319723280. ISSN 16113349. Disponível em: <https://link.springer.com/chapter/10.1007/978-3-319-72329-7_18>.

IEEE. Proceedings of Conference on Intelligent Transportation Systems. In: *Proceedings of Conference on Intelligent Transportation Systems ITSC-97*. Boston, MA, USA: IEEE, 1997.

JANNACH, D.; ZANKER, M.; FUCHS, M. Constraint-based recommendation in tourism: A multiperspective case study. *Information Technology & Tourism*, Cognizant Communication Corporation, v. 11, n. 2, p. 139–155, 2009. ISSN 1098-3058.

JIANG, J.; WANG, C.; CHATTOPADHYAY, S.; ZHANG, W. Road Context-Aware Intrusion Detection System for Autonomous Cars. In: ZHOU, J.; LUO, X.; SHEN, Q.; XU, Z. (Ed.). *Lecture Notes in Computer Science*. Cham, Switzerland: Springer Nature, 2020. v. 11999.

JOHNSON, D. A.; TRIVEDI, M. M. Driving style recognition using a smartphone as a sensor platform. In: *14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. [S.l.]: IEEE, 2011. p. 1609–1615. ISBN 978-1-4577-2197-7.

KALTZ, W. J.; ZIEGLER, J.; LOHMANN, S. Context-aware Web engineering: Modeling and applications. *Revue d'intelligence artificielle*, v. 19, n. 3, p. 439–458, 2005.

KANNAN, S.; THANGAVELU, A.; KALIVARADHAN, R. An intelligent Driver Assistance System (I-DAS) for vehicle safety modelling using ontology approach. *International Journal of UbiComp*, v. 1, n. 3, p. 15–29, 2010.

KARIMI, M.; JANNACH, D.; JUGOVAC, M. News recommender systems – Survey and roads ahead. *Information Processing and Management*, Elsevier Ltd, v. 54, n. 6, p. 1203–1227, 11 2018. ISSN 03064573.

KHEKARE, G. S.; SAKHARE, A. V. Intelligent traffic system for VANET: a survey. *International Journal of Advanced Computer Research*, Accent Social and Welfare Society, v. 2, n. 6, p. 99–102, 12 2012.

KLOTZ, B.; TRONCY, R.; WILMS, D.; BONNET, C. VSSo-A Vehicle Signal and Attribute Ontology. In: *SSN Workshop at ISWC. CEUR Workshop Proceedings*. [S.l.: s.n.], 2018.

KOLBE, N.; KUBLER, S.; ROBERT, J.; TRAON, Y. L.; ZASLAVSKY, A. Towards semantic interoperability in an open IoT ecosystem for connected vehicle services. In: *GIoTS 2017 - Global Internet of Things Summit, Proceedings*. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2017. ISBN 9781509058730.

KOREN, Y.; BELL, R. Advances in Collaborative Filtering. *Recommender Systems Handbook, Second Edition*, Springer, Boston, MA, p. 77–118, 1 2015. Disponível em: <https://link.springer.com/chapter/10.1007/978-1-4899-7637-6_3>.

LATHIA, N.; HAILES, S.; CAPRA, L. The effect of correlation coefficients on communities of recommenders. *Proceedings of the ACM Symposium on Applied Computing*, p. 2000–2005, 2008.

LEYVAND, T.; MEEKHOF, C.; Yi-Chen Wei; Jian Sun; Baining Guo. Kinect Identity: Technology and Experience. *Computer*, v. 44, n. 4, p. 94–96, 4 2011.

LI, B.; WANG, Y.; SINGH, A.; VOROBEYCHIK, Y. Data Poisoning Attacks on Factorization-Based Collaborative Filtering. In: LEE, D. D.; SUGIYAMA, M.; LUXBURG, U. V.; GUYON, I.; GARNETT, R. (Ed.). *Advances in Neural Information Processing Systems*. Barcelona: Curran Associates, Inc, 2016. v. 29, p. 1893–1901. ISBN 978-1-5108-3881-9.

LIN, K. P.; SHEN, C. Y.; CHANG, T. L.; CHANG, T. M. A Consumer Review-Driven Recommender Service for Web E-Commerce. *Proceedings - 2017 IEEE 10th International Conference on Service-Oriented Computing and Applications, SOCA 2017*, Institute of Electrical and Electronics Engineers Inc., v. 2017-January, p. 206–210, 12 2017.

LOHMANN, S.; NEGRU, S.; HAAG, F.; ERTL, T. Visualizing ontologies with VOWL. *Semantic Web*, IOS Press, v. 7, n. 4, p. 399–419, 1 2016. ISSN 1570-0844.

MASLEKAR, N.; BOUSSEDJRA, M.; MOUZNA, J.; LABIOD, H. VANET based adaptive traffic signal control. In: *IEEE Vehicular Technology Conference*. [S.l.: s.n.], 2011. ISBN 9781424483310. ISSN 15502252.

MEIER, R.; HARRINGTON, A.; CAHILL, V. Towards delivering context-aware transportation user services. In: *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2006. p. 369–376. ISBN 1424400945.

MENSAH, I. K.; MWAKAPESA, D. S. The Impact of Context Awareness and Ubiquity on Mobile Government Service Adoption. *Mobile Information Systems*, Hindawi Limited, v. 2022, 2022. ISSN 1875905X.

MITCHELL, M.; MEYERS, C.; WANG, A.-I. A.; TYSON, G. Contextprovider: Context awareness for medical monitoring applications. In: *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*. [S.l.: s.n.], 2011. p. 5244–5247.

MONDAL, M. A.; REHENA, Z. An IoT-Based Congestion Control Framework for Intelligent Traffic Management System. *Advances in Intelligent Systems and Computing*, Springer, Singapore, v. 1133, p. 1287–1297, 2021. Disponível em: <https://link.springer.com/chapter/10.1007/978-981-15-3514-7_96>.

MOUGOUEI, D.; POWERS, D. M. Dependency-aware software requirements selection using fuzzy graphs and integer programming. *Expert Systems with Applications*, Elsevier Ltd, v. 167, p. 113748–undefined, 2021. ISSN 09574174.

NAKAMURA, H.; GAO, Y.; GAO, H.; ZHANG, H.; KIYOHIRO, A.; MINE, T. Adaptive user interface agent for personalized public transportation recommendation system: PATRASH. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer Verlag, 2014. v. 8861, p. 238–245. ISBN 9783319131900. ISSN 16113349. Disponível em: <https://link.springer.com/chapter/10.1007/978-3-319-13191-7_19>.

NARAYANAN, V.; REHMAN, R.; DEVASSY, A.; RAMA, S.; AHLUWALIA, P.; RAMACHANDRAN, A. Enabling location based services for hyperlocal marketing in connected vehicles. In: *2014 International Conference on Connected Vehicles and Expo, ICCVE 2014 - Proceedings*. Vienna, Austria: Institute of Electrical and Electronics Engineers Inc., 2014. p. 12–13. ISBN 9781479967292.

NASSAR, L.; KAMEL, M. S.; KARRAY, F. VANET IR-CAS for Safety ACN: Information Retrieval Context Aware System for VANET Automatic Crash Notification Safety Application. *International Journal of Intelligent Transportation Systems Research*, Springer New York LLC, v. 14, n. 3, p. 127–138, 9 2016. ISSN 18688659. Disponível em: <https://link.springer.com/article/10.1007/s13177-014-0108-x>.

NASUTION, M. K. M.; Nasution; M., M. K. A method for constructing a dataset to reveal the industrial behaviour of big data. *MS&E*, IOP Publishing Ltd, v. 1003, n. 1, p. 012156, 12 2020. ISSN 1757899X. Disponível em: <https://ui.adsabs.harvard.edu/abs/2020MS&E.1003a2156N/abstract>.

NGAI, E. W. T.; LEUNG, T. K. P.; WONG, Y. H.; LEE, M. C. M.; CHAI, P. Y. F.; CHOI, Y. S. Design and development of a context-aware decision support system for real-time accident handling in logistics. *Decision support systems*, Elsevier, v. 52, n. 4, p. 816–827, 2012. ISSN 0167-9236.

ÖZKUL, M.; CAPUNI, I.; DOMNORI, E. Context-aware intelligent traffic light control through secure messaging. *Journal of Advanced Transportation*, Hindawi Limited, v. 2018, 2018.

PANAGIOTOPOULOS, I.; DIMITRAKOPOULOS, G. Cognitive Infotainment Systems for Intelligent Vehicles. In: *10th International Conference on Information, Intelligence, Systems and Applications, IISA 2019*. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2019. ISBN 9781728149592.

Pandora Media Inc. *Pandora: Music and Podcasts, Free and On-Demand*. 2022.

PAPATHEOCHAROUS, E.; WNUK, K.; PETERSEN, K.; SENTILLES, S.; CICCHETTI, A.; GORSCHEK, T.; SHAH, S. M. A. The GRADE taxonomy for supporting decision-making of asset selection in software-intensive system development. *Information and Software Technology*, Elsevier, v. 100, p. 1–17, 8 2018. ISSN 0950-5849.

PARODI, A.; MARESCA, M.; PROVERA, M.; BAGLIETTO, P. An IoT approach for the connected vehicle. In: *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*. Springer Verlag, 2016. v. 170, p. 158–161. ISBN 9783319470740. ISSN 18678211. Disponível em: <https://link.springer.com/chapter/10.1007/978-3-319-47075-7_19>.

PARUNDEKAR, R.; OGUCHI, K. Learning driver preferences of POIs using a semantic web knowledge system. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer, Berlin, Heidelberg, 2012. v. 7295 LNCS, p. 703–717. ISBN 9783642302831. ISSN 03029743. Disponível em: <https://link.springer.com/chapter/10.1007/978-3-642-30284-8_54>.

PEISCHL, B.; NICA, M.; ZANKER, M.; SCHMID, W. Recommending effort estimation methods for software project management. In: *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*. [S.l.]: IEEE, 2009. v. 3, p. 77–80. ISBN 0769538010.

QUIJANO-SÁNCHEZ, L.; CANTADOR, I.; CORTÉS-CEDIEL, M. E.; GIL, O. *Recommender systems for smart cities*. [S.l.]: Elsevier Ltd, 2020.

RAJARAM, H. K.; LOANE, J.; MACMAHON, S. T.; CAFFERY, F. M. Taxonomy-based testing and validation of a new defect classification for health software. *Journal of Software: Evolution and Process*, John Wiley & Sons, Ltd, v. 31, n. 1, p. e1985, 1 2019. ISSN 2047-7481. Disponível em: <https://onlinelibrary.wiley.com/doi/full/10.1002/smr.1985https://onlinelibrary.wiley.com/doi/abs/10.1002/smr.1985https://onlinelibrary.wiley.com/doi/10.1002/smr.1985>.

RAMAZANI, A.; VAHDAT-NEJAD, H. A new context-aware approach to traffic congestion estimation. In: *Proceedings of the 4th International Conference on Computer and Knowledge Engineering, ICCKE 2014*. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2014. p. 504–508. ISBN 9781479954865.

RAMESH, M. V.; VIDYA, P. T.; PRADEEP, P. Context aware wireless sensor system integrated with participatory sensing for real time road accident detection. In: *IFIP International Conference on Wireless and Optical Communications Networks, WOCN*. Bhopal, India: [s.n.], 2013. ISBN 9781467359993. ISSN 21517681.

RAPHIPHAN, P.; ZASLAVSKY, A.; PRATHOMBUTR, P.; MEESAD, P. Context aware traffic congestion estimation to compensate intermittently available mobile sensors. In: *Proceedings - IEEE International Conference on Mobile Data Management*. [S.l.: s.n.], 2009. p. 405–410. ISBN 9780769536507. ISSN 15516245.

RAUSCHER, S.; MESSNER, G.; BAUR, P.; AUGENSTEIN, J.; DIGGES, K. H.; PERDECK, E. Enhanced Automatic Collision Notification (ACN) System - Improved Rescue Care Due to Injury Prediction - First Field Experience. In: *Proceedings of the 21st International Technical Conference on the Enhanced Safety of Vehicles (ESV 2009)*. Stuttgart, Germany: [s.n.], 2009. p. 9–49.

REIS, T.; DUARTE, C.; CARRIÇO, L.; CARVALHO, R. Towards a Context Aware Multimodal Hand-Held Device. In: RODRIGUES, L.; LOPES, R. (Ed.). *Actas do INForum - Simpósio de Informática 2009*. Lisbon, Portugal: Faculdade de Ciencias da Universidade de Lisboa, 2009. v. 2009, p. 239–250.

RICCI, F.; CAVADA, D.; MIRZADEH, N.; VENTURINI, A. Case-based travel recommendations. *Destination recommendation systems: behavioural foundations and applications*, p. 67–93, 2006.

RICCI, F.; ROKACH, L.; SHAPIRA, B. Introduction to Recommender Systems Handbook. In: *Recommender Systems Handbook*. New York: Springer US, 2011. cap. 1, p. 1–35. ISBN 978-0-387-85819-7. Disponível em: <https://link.springer.com/chapter/10.1007/978-0-387-85820-3_1>.

RICO, J.; SANCHO, J.; CENDON, B.; CAMUS, M. Parking easier by using context information of a smart city: Enabling fast search and management of parking resources. In: *Proceedings - 27th International Conference on Advanced Information Networking and Applications Workshops, WAINA 2013*. Barcelona, Spain: [s.n.], 2013. p. 1380–1385. ISBN 9780769549521.

ROCHA, N. P.; DIAS, A.; SANTINHA, G.; RODRIGUES, M.; RODRIGUES, C.; QUEIRÓS, A.; BASTARDO, R.; PAVÃO, J. Systematic literature review of context-awareness applications supported by smart cities' infrastructures. *SN Applied Sciences 2022 4:4*, Springer, v. 4, n. 4, p. 1–19, 3 2022. ISSN 2523-3971. Disponível em: <https://link.springer.com/article/10.1007/s42452-022-04979-0>.

SABET, A. J.; ROSSI, M.; SCHREIBER, F. A.; TANCA, L. Towards Learning Travelers' Preferences in a Context-Aware Fashion. *Advances in Intelligent Systems and Computing*, Springer, Cham, v. 1239 AISC, p. 203–212, 6 2020. Disponível em: <https://link.springer.com/chapter/10.1007/978-3-030-58356-9_20>.

SADOUN, B.; AL-BAYARI, O. Location based services using geographical information systems. *Computer Communications*, v. 30, n. 16, p. 3154–3160, 11 2007. ISSN 01403664. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S0140366407002629>.

SAHA, A.; CHAKI, R. VAMI - A novel architecture for vehicular ambient intelligent system. In: *Communications in Computer and Information Science*. Springer, Berlin, Heidelberg, 2011. v. 245 CCIS, p. 55–64. ISBN 9783642272448. ISSN 18650929. Disponível em: <https://link.springer.com/chapter/10.1007/978-3-642-27245-5_9>.

SANTA, J.; GÓMEZ-SKARMETA, A. F. Sharing context-aware road and safety information. *IEEE Pervasive Computing*, v. 8, n. 3, p. 58–65, 7 2009. ISSN 15361268.

SANTOS, G.; NIKOLAEV, N. Mobility as a Service and Public Transport: A Rapid Literature Review and the Case of Moovit. *Sustainability 2021, Vol. 13, Page 3666*, Multidisciplinary Digital Publishing Institute, v. 13, n. 7, p. 3666, 3 2021. Disponível em: <https://www.mdpi.com/2071-1050/13/7/3666/htmhttps://www.mdpi.com/2071-1050/13/7/3666>.

SCHRAGE, M. *Recommendation Engines*. [S.l.]: MIT Press, 2020. ISBN 0262358786.

SEPULCRE, M.; GOZALVEZ, J. Context-aware heterogeneous V2X communications for connected vehicles. *Computer Networks*, Elsevier, v. 136, p. 13–21, 2018. ISSN 1389-1286.

SEREDYNSKI, M.; ARNOULD, G.; KHADRAOUI, D. The Emerging Applications of Intelligent Vehicular Networks for Traffic Efficiency. In: *Proceedings of the Third ACM International Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications*. New York, NY, USA: ACM, 2013. (DIVANet '13), p. 101–108. ISBN 978-1-4503-2358-1.

SHESHASAAYEE, A.; MUNIYANDI, P. Recommender Systems: An Introduction and Overview. *The International Journal of Analytical and Experimental Modal Analysis*, XII, n. I, p. 1594–1599, 1 2020. ISSN 0886-9367.

SILVA, A. V. D. da; BORGES, L.; VIEIRA, V. CDNA - A context-aware notification system for driver interruption. In: *IHC 2018: Proceedings of the 17th Brazilian Symposium on Human Factors in Computing Systems*. Belém, Brazil: Association for Computing Machinery, 2018. p. 1–8. ISBN 9781450366014. Disponível em: <http://dl.acm.org/citation.cfm?doid=3274192.3274203>.

SINHA, A. P.; POPKEN, D. Completeness and consistency checking of system requirements: An expert agent approach. *Expert Systems with Applications*, Pergamon, v. 11, n. 3, p. 263–276, 1 1996. ISSN 09574174.

SMITH, B.; LINDEN, G. Two Decades of Recommender Systems at Amazon.com. *IEEE Internet Computing*, Institute of Electrical and Electronics Engineers Inc., v. 21, n. 3, p. 12–18, 5 2017. ISSN 10897801.

SOVACOOL, B. K.; ROGGE, J. C.; SALETA, C.; MASTERSON-COX, E. Transformative versus conservative automotive innovation styles: Contrasting the electric vehicle manufacturing strategies for the BMW i3 and Fiat 500e. *Environmental Innovation and Societal Transitions*, Elsevier B.V., v. 33, p. 45–60, 11 2019. ISSN 22104224.

SOYTURK, M.; MUHAMMAD, K. N.; AVCIL, M. N.; KANTARCI, B.; MATTHEWS, J. Chapter 8 - From vehicular networks to vehicular clouds in smart cities. In: OBAIDAT, M. S.; NICOPOLITIDIS, P. (Ed.). *Smart Cities and Homes*. Boston: Morgan Kaufmann, 2016. p. 149 – 171. ISBN 978-0-12-803454-5. Disponível em: <http://www.sciencedirect.com/science/article/pii/B9780128034545000080>.

SRIDEVI, M.; RAO, R. R.; RAO, V. M. A Survey on Recommender System. *IJCSIS) International Journal of Computer Science and Information Security*, v. 14, n. 5, p. 265–272, 5 2016. ISSN 1947-5500. Disponível em: <https://sites.google.com/site/ijcsis/>.

SUBRAMANYAM, M.; KUMAR, K. N. A. RLTS: Recommendation for local transportation system using Ambient Intelligence. In: *2015 International Conference on Emerging Research in Electronics, Computer Science and Technology, ICERECT 2015*. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2016. p. 65–69. ISBN 9781467395632.

SUJITHA, T.; PUNITHA, D. S. Intelligent transportation system for vehicular ad-hoc networks. *International Journal Of Technology Enhancements And Emerging Engineering Research*, v. 2, n. 6, p. 69–73, 2014. ISSN 2347-4289.

SUKODE, S.; GITE, S. Vehicle traffic congestion control & monitoring system in IoT. *International Journal of Applied Engineering Research*, v. 10, n. 8, p. 19513–19523, 2015.

SWARNAMUGI, M.; CHINNAIYAN, R. Context—Aware Smart Reliable Service Model for Intelligent Transportation System Based on Ontology. *Lecture Notes in Electrical Engineering*, Springer, Cham, v. 597, p. 23–30, 2020. Disponível em: <https://link.springer.com/chapter/10.1007/978-3-030-29407-6_3>.

TANG, Z.; JAYAKAR, K.; FENG, X.; ZHANG, H.; PENG, R. X. Identifying smart city archetypes from the bottom up: A content analysis of municipal plans. *Telecommunications*

Policy, v. 43, n. 10, p. 101834, 2019. ISSN 0308-5961. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0308596118303367>.

TAO, C.; JIANG, Q.; DUAN, L.; LUO, P. Dynamic and Static Context-Aware LSTM for Multi-agent Motion Prediction. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer, Cham, v. 12366 LNCS, p. 547–563, 8 2020. Disponível em: <https://link.springer.com/chapter/10.1007/978-3-030-58589-1_33>.

TEMDEE, P.; PRASAD, R. Context and Its Awareness. In: *Context-Aware Communication and Computing: Applications for Smart Environment*. Cham, Switzerland: Springer, 2018. p. 15–31.

THANH-TAI, H.; NGUYEN, H.-H.; THAI-NGHE, N. A semantic approach in recommender systems. In: *International Conference on Future Data and Security Engineering*. [S.l.]: Springer, 2016. p. 331–343.

TINTAREV, N.; MASTHOFF, J. Designing and Evaluating Explanations for Recommender Systems. In: RICCI, F.; LIOR, R.; SHAPIRA, B.; KANTOR, P. B. (Ed.). *Recommender Systems Handbook*. New York: Springer US, 2011. cap. 15, p. 479–510. ISBN 978-0-387-85819-7. Disponível em: <https://link.springer.com/chapter/10.1007/978-0-387-85820-3_15>.

VAHDAT-NEJAD, H.; RAMAZANI, A.; MOHAMMADI, T.; MANSOOR, W. A survey on context-aware vehicular network applications. *Vehicular Communications*, Elsevier BV, v. 3, n. C, p. 43–57, 1 2016.

VIEIRA, V.; CALDAS, L. R.; SALGADO, A. C. Towards an Ubiquitous and Context Sensitive Public Transportation System. In: *2011 Fourth International Conference on Ubi-Media Computing*. São Paulo, Brazil: IEEE, 2011. p. 174–179. ISBN 978-1-4577-1174-9.

VIEIRA, V.; TEDESCO, P.; SALGADO, A. C. Towards an ontology for context representation in groupware. In: *11th International Workshop, CRIWG*. Porto de Galinhas, Brazil: [s.n.], 2005. v. 3706, p. 367–375.

VIEIRA, V.; TEDESCO, P.; SALGADO, A. C. A process for the design of Context-Sensitive Systems. In: *2009 13th International Conference on Computer Supported Cooperative Work in Design*. Santiago, Chile: IEEE, 2009. p. 143–148.

VIEIRA, V.; TEDESCO, P.; SALGADO, A. C.; BRÉZILLON, P. Investigating the Specifics of Contextual Elements Management: The CEManTIKA Approach. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer, Berlin, Heidelberg, v. 4635 LNAI, p. 493–506, 2007. ISSN 16113349. Disponível em: <https://link.springer.com/chapter/10.1007/978-3-540-74255-5_37>.

VIJAYMEENA, M. K.; KAVITHA, K. A survey on similarity measures in text mining. *Machine Learning and Applications: An International Journal*, v. 3, n. 2, p. 19–28, 2016.

VILLEGAS, N. M.; MÜLLER, H. A.; MUÑOZ, J. C.; LAU, A.; NG, J.; BREALEY, C. A dynamic context management infrastructure for supporting user-driven web integration in the personal web. In: *Proceedings of the 2011 Conference of the Center for Advanced Studies on Collaborative Research*. Toronto, Canada: IBM Corp. Riverton, NJ, USA, 2011. p. 200–214.

WANG, C.; DAVID, B.; CHALON, R. A smart city case study: Dynamic management of road lanes. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer Verlag, 2014. v. 8530 LNCS, p. 629–640. ISBN 9783319077871. ISSN 16113349. Disponível em: <https://link.springer.com/chapter/10.1007/978-3-319-07788-8_58>.

WANG, Y.; JIANG, J.; MU, T. Context-aware and energy-driven route optimization for fully electric vehicles via crowdsourcing. *IEEE Transactions on Intelligent Transportation Systems*, v. 14, n. 3, p. 1331–1345, 2013. ISSN 15249050.

WATERMAN, D. A. *A guide to expert systems*. 1. ed. United States: Addison-Wesley Longman Publishing Company, 1986. 419 p. ISBN 0201083132.

WERTHER, B.; HOCH, N. E-mobility as a challenge for new ICT solutions in the car industry. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer, Berlin, Heidelberg, 2012. v. 7173 LNCS, p. 46–57. ISBN 9783642300646. ISSN 03029743. Disponível em: <https://link.springer.com/chapter/10.1007/978-3-642-30065-3_3>.

WINDER, C.; AZZI, R.; WAGNER, D. The development of the globally harmonized system (GHS) of classification and labelling of hazardous chemicals. *Journal of Hazardous Materials*, Elsevier, v. 125, n. 1-3, p. 29–44, 10 2005. ISSN 0304-3894.

WOERND, W.; EIGNER, R. Collaborative, context-aware applications for inter-networked cars. In: *Proceedings of the Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE*. Paris, France: [s.n.], 2007. p. 180–185. ISBN 0769528791. ISSN 15244547.

WONGCHAROEN, S.; SENIVONGSE, T. Twitter analysis of road traffic congestion severity estimation. *2016 13th International Joint Conference on Computer Science and Software Engineering, JCSSE 2016*, Institute of Electrical and Electronics Engineers Inc., 11 2016.

WOODARD, M.; WISELY, M.; SARVESTANI, S. S. A Survey of Data Cleansing Techniques for Cyber-Physical Critical Infrastructure Systems. *Advances in Computers*, Elsevier, v. 102, p. 63–110, 1 2016. ISSN 0065-2458.

XIONG, Z.; DIXIT, V. V.; WALLER, S. T. The development of an Ontology for driving Context Modelling and reasoning. In: *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. [S.l.: s.n.], 2016. p. 13–18. ISBN 2153-0017.

YOUNES, M. B.; BOUKERCHE, A.; MAMMERI, A. Context-Aware traffic light self-scheduling algorithm for intelligent transportation systems. In: *IEEE Wireless Communications and Networking Conference, WCNC*. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2016. v. 2016-September, p. 1–6. ISBN 9781467398145. ISSN 15253511.

ZARDOSHT, B.; BEAUCHEMIN, S.; BAUER, M. A. An in-vehicle tracking method using vehicular Ad-Hoc networks with a vision-based system. In: *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*. San Diego, CA, USA: Institute of Electrical and Electronics Engineers Inc., 2014. p. 3022–3029. ISSN 1062922X.

ZARZA, H.; KARIMI, H.; MOHAMMAD, S.; HOSSEINI, S. Y.; ATEAIAN, H. R.; ROSTAMI, H. CAGFP: Context Aware Geocast Forwarding Protocol for Inter-Vehicle Communication. In: *21st Iranian Conference on Electrical Engineering (ICEE)*. Mashhad, Iran: IEEE, 2013.

ZHANG, W.; CHENG, B.; LIN, Y. Driver drowsiness recognition based on computer vision technology. *Tsinghua Science and Technology*, v. 17, n. 3, p. 354–362, 2012. ISSN 10070214.

ZHENG, X.; CHEN, W.; WANG, P.; SHEN, D.; CHEN, S.; WANG, X.; ZHANG, Q.; YANG, L. Big Data for Social Transportation. *IEEE Transactions on Intelligent Transportation Systems*, Institute of Electrical and Electronics Engineers (IEEE), v. 17, n. 3, p. 620–630, 3 2016.

ZIARANI, R. J.; RAVANMEHR, R. Deep neural network approach for a serendipity-oriented recommendation system. *Expert Systems with Applications*, Pergamon, v. 185, p. 115660, 12 2021. ISSN 0957-4174.

ZIMMERMANN, A.; LORENZ, A.; OPPERMANN, R. An Operational Definition of Context. In: *Modeling and Using Context*. [S.l.]: Springer Berlin Heidelberg, 2007. p. 558–571.

APPENDIX A – KNOWLEDGE BASE 1.0

Frame 7 – Knowledge base of Projects x Context Element Categories - Version 1.0 without the four projects unfit for use in the recommender engine

Context element	Goto and Kambayashi (2002)	Gena and Torre (2004)	Sadoun and Al-Bayari (2007)	Saha and Chaki (2011)	Werther and Hoch (2012)	Parundekar and Oguchi (2012)	David et al. (2013) - Communicating Bus Stop	David et al. (2013) - Loading Zone	Wang et al. (2014)	Narayanan et al. (2014)	Nakamura et al. (2014)	Subramanyam and Ashwath Kumar (2016)	Parodi et al. (2016)	Kolbe et al. (2017)	Hu et al. (2017)	Atasoy et al. (2015)	Figueiredo et al. (2001)	Panagiotopoulos and Dimitrakopoulos (2019)	Santa and Gómez-Skarmeta (2009)	Arkian et al. (2014)	Barba et al. (2013)	Bifulco et al. (2014) - Amsterdam	Bifulco et al. (2014) - Singapore	Rauscher et al. (2009)	Nassar et al. (2016)	Ramazani and Vahdat-Nejad (2014)	Rico et al. (2013)	Raphiphan et al. (2009)	Baltrunas et al. (2011)	Ngai et al. (2012)	Hoogendoorn et al. (2012)	Alghamdi (2012)	Woernd and Eigner (2007)	Fuchs et al. (2008)	Wang et al. (2013)	Zarza et al. (2013)	Al-Sultan et al. (2013)	Ramesh et al. (2013)	Alghamdi et al. (2012)	Bae and Olariu (2010)	Alhammad et al. (2012)	Bergan et al. (1998)	Alazawi et al. (2011)	Elhadef (2015)	Ghaffarian et al. (2012)	Fogue et al. (2011)	Maslekar et al. (2011)	Khekare and Sakhare (2012)	Zardosht et al. (2014)	Sujitha and Punitha (2014)	Sukode and Gite (2015)	Younes et al. (2016)	Aguirre et al. (2017)	Meier et al. (2006)	Kannan et al. (2010)	Vieira et al. (2011)	da Silva et al. (2018)	Supra-categories
Driver	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	Activity
Driving Task	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	
Driver External Task	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Driver Infotainment task	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Road Availability	0	1	0	1	0	0	0	1	1	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	
Traffic Condition	0	0	0	0	1	0	0	1	1	0	0	0	1	0	0	0	1	1	0	1	0	0	0	0	0	1	1	1	1	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	1	1	0	1	0	1	0	
Passenger	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Cargo	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Pedestrian	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	
Pedestrian Intention	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Pedestrian Acceleration	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Pedestrian Direction	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Pedestrian Speed	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Pedestrian Role	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Social Media	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Surroundings Climate	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0	
Surroundings RSU / Ext. Infrastructure	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	
Surroundings Road	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
Surroundings Traffic Lights	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	1	0	0	1	1	0	0	0	0	0	
Surroundings Traffic Sign	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Surroundings Person Cyclist	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	
Surroundings Person Driver	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Surroundings Person Passenger	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	
Surroundings Person Pedestrian	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	
Surroundings Person Traffic Controller	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Surroundings Person Traffic Police	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Surroundings Point of Interest	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Surroundings Vehicle	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0	0	1	0	1	0	1	1	1	1	1	1	0	0	1	0	0	1	0	1	1	0	0	0	0	1	0	0	
Network	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	1	0	0	0	0	
Connectivity	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Peer Device Infrastructure	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Peer Output Device	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Peer Sensor	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	
Peer Vehicle	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	
Vehicle	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	1	0	0	
Mechanical Status	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1	0	1	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	
Vehicle Movement	0	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	1	0	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	1	0	0	1	1	1	0	1	0	0	0	1	0	1	1	1	
Road Characteristics	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	Identity
Device Type	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Device UID	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	
Vehicle Id	0	0	0	0	1	0	0	0	1	0	0	1	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	1	0	1	1	1	0	0	1	1	0	0	1	1	0	1	0	
Vehicle Dimensions	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	
Vehicle Mechanical Attributes	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	
Vehicle Type	0	0	1	0	0	0	0	0	1	0	0	1	0	1	1	1	0	0	0	0	0	1	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1	1	1	1	1	0	
Place Id	1	0	0	1	1	1	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	
Place Attributes	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Place UID	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Payload Id	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Cargo Identification	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Payload Dimension	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Payload Type	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Passenger Id	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	
Person Id	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Driver Id	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Profile	1	1	1	0	1	1	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1	0	1	0	1	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Driver Experience	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Person UID - Knowledge	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Person UID - Physical Attribute	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Person UID - Possession	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Physical Attributes	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	
Person Statistics	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	
Geographic Coordinate	0	1	1	1	0	1	1	0	0	1	0	0	1	1	0	1	0	0	1	1	1	0	0	1	0	0	1	0	1	1	1	1	1	0	0	1	1	1	0	0	1	0	0	1	1	1	1	1	1	0	0	0	0	1	0	1	0	Location
Geographic Location	1	0	0	0	0	1	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	1	0	0	0	0	0	0	1	0	0	0	
Location Type	1	0	0	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	
Route	0	0	0	0	1	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	
Symbolic Coordinate	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0	
Distance Traveled	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
Time	1	1	0	0	1	1	0	1	0	0	1	0	0	0	1	1	1	1	0	0	1	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	1	0	0	1	1	0	0	0	1	0	1	0	Time
Schedule	1	0	0	0	1	0	0	1	0	0	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	
Travel Time	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Source: The author (2022)

APPENDIX B – COSINE SIMILARITY MATRIX

Frame 8 – Cosine similarity matrix

	da Silva et al. (2018)	Vieira et al. (2011)	Kannan et al. (2010)	Meier et al. (2006)	Aguirre et al. (2017)	Younes et al. (2016)	Sukode and Gite (2015)	Sujitha and Punitha (2014)	Zardosht et al. (2014)	Khekare and Sakhare (2012)	Maslekar et al. (2011)	Fogue et al. (2011)	Ghaffarian et al. (2012)	Elhadef (2015)	Alazawi et al. (2011)	Bergan et al. (1998)	Alhammad et al. (2012)	Bae and Olariu (2010)	Alghamdi et al. (2012)
da Silva et al. (2018)	1.000	0.183	0.320	0.000	0.183	0.000	0.000	0.000	0.192	0.000	0.236	0.174	0.258	0.000	0.000	0.236	0.000	0.333	0.218
Vieira et al. (2011)	0.183	1.000	0.175	0.730	0.300	0.316	0.183	0.258	0.527	0.365	0.387	0.477	0.424	0.258	0.286	0.387	0.191	0.183	0.359
Kannan et al. (2010)	0.320	0.175	1.000	0.160	0.526	0.277	0.160	0.113	0.277	0.000	0.226	0.334	0.124	0.226	0.000	0.340	0.084	0.480	0.419
Meier et al. (2006)	0.000	0.730	0.160	1.000	0.274	0.289	0.333	0.236	0.481	0.333	0.236	0.348	0.258	0.354	0.348	0.236	0.348	0.167	0.327
Aguirre et al. (2017)	0.183	0.300	0.526	0.274	1.000	0.316	0.183	0.258	0.316	0.000	0.129	0.381	0.283	0.129	0.095	0.387	0.095	0.365	0.359
Younes et al. (2016)	0.000	0.316	0.277	0.289	0.316	1.000	0.577	0.000	0.000	0.577	0.204	0.302	0.000	0.000	0.302	0.204	0.000	0.000	0.189
Sukode and Gite (2015)	0.000	0.183	0.160	0.333	0.183	0.577	1.000	0.000	0.000	0.667	0.236	0.000	0.000	0.000	0.348	0.000	0.000	0.333	0.436
Sujitha and Punitha (2014)	0.000	0.258	0.113	0.236	0.258	0.000	0.000	1.000	0.544	0.000	0.167	0.246	0.183	0.333	0.246	0.333	0.246	0.236	0.309
Zardosht et al. (2014)	0.192	0.527	0.277	0.481	0.316	0.000	0.000	0.544	1.000	0.192	0.408	0.402	0.447	0.544	0.302	0.272	0.402	0.385	0.504
Khekare and Sakhare (2012)	0.000	0.365	0.000	0.333	0.000	0.577	0.667	0.000	0.192	1.000	0.471	0.174	0.258	0.236	0.348	0.000	0.174	0.000	0.218
Maslekar et al. (2011)	0.236	0.387	0.226	0.236	0.129	0.204	0.236	0.167	0.408	0.471	1.000	0.246	0.365	0.500	0.369	0.167	0.369	0.471	0.463
Fogue et al. (2011)	0.174	0.477	0.334	0.348	0.381	0.302	0.000	0.246	0.402	0.174	0.246	1.000	0.405	0.246	0.091	0.369	0.182	0.174	0.228
Ghaffarian et al. (2012)	0.258	0.424	0.124	0.258	0.283	0.000	0.000	0.183	0.447	0.258	0.365	0.405	1.000	0.365	0.000	0.365	0.270	0.258	0.169
Elhadef (2015)	0.000	0.258	0.226	0.354	0.129	0.000	0.000	0.333	0.544	0.236	0.500	0.246	0.365	1.000	0.123	0.167	0.492	0.236	0.154
Alazawi et al. (2011)	0.000	0.286	0.000	0.348	0.095	0.302	0.348	0.246	0.302	0.348	0.369	0.091	0.000	0.123	1.000	0.000	0.273	0.000	0.228
Bergan et al. (1998)	0.236	0.387	0.340	0.236	0.387	0.204	0.000	0.333	0.272	0.000	0.167	0.369	0.365	0.167	0.000	1.000	0.123	0.236	0.154
Alhammad et al. (2012)	0.000	0.191	0.084	0.348	0.095	0.000	0.000	0.246	0.402	0.174	0.369	0.182	0.270	0.492	0.273	0.123	1.000	0.174	0.114
Bae and Olariu (2010)	0.333	0.183	0.480	0.167	0.365	0.000	0.333	0.236	0.385	0.000	0.471	0.174	0.258	0.236	0.000	0.236	0.174	1.000	0.655
Alghamdi et al. (2012)	0.218	0.359	0.419	0.327	0.359	0.189	0.436	0.309	0.504	0.218	0.463	0.228	0.169	0.154	0.228	0.154	0.114	0.655	1.000
Ramesh et al. (2013)	0.218	0.359	0.419	0.327	0.478	0.000	0.218	0.309	0.504	0.218	0.463	0.342	0.507	0.463	0.000	0.309	0.342	0.655	0.571
Al-Sultan et al. (2013)	0.365	0.500	0.526	0.365	0.200	0.158	0.365	0.258	0.527	0.365	0.516	0.286	0.283	0.387	0.286	0.129	0.191	0.548	0.478
Zarza et al. (2013)	0.289	0.316	0.277	0.144	0.158	0.000	0.000	0.204	0.500	0.289	0.612	0.302	0.671	0.408	0.000	0.204	0.302	0.577	0.378
Wang et al. (2013)	0.192	0.316	0.370	0.289	0.316	0.167	0.385	0.136	0.222	0.192	0.136	0.101	0.149	0.000	0.101	0.408	0.101	0.385	0.504
Fuchs et al. (2008)	0.408	0.112	0.588	0.000	0.224	0.177	0.000	0.289	0.236	0.000	0.289	0.213	0.158	0.289	0.000	0.433	0.107	0.408	0.267
Woernd and Eigner (2007)	0.183	0.300	0.263	0.365	0.200	0.000	0.000	0.000	0.422	0.183	0.258	0.286	0.283	0.129	0.191	0.129	0.477	0.183	0.359
Alghamdi (2012)	0.289	0.316	0.277	0.144	0.158	0.000	0.000	0.204	0.500	0.289	0.612	0.302	0.447	0.408	0.000	0.408	0.302	0.577	0.378
Hoogendoorn et al. (2012)	0.577	0.316	0.277	0.144	0.316	0.000	0.000	0.000	0.333	0.289	0.408	0.302	0.447	0.204	0.000	0.204	0.151	0.289	0.378
Ngai et al. (2012)	0.000	0.456	0.240	0.417	0.274	0.144	0.000	0.118	0.289	0.167	0.118	0.261	0.258	0.236	0.000	0.354	0.435	0.000	0.218
Baltrunas et al. (2011)	0.192	0.316	0.277	0.385	0.105	0.167	0.385	0.272	0.222	0.385	0.136	0.302	0.149	0.136	0.201	0.136	0.201	0.192	0.252
Raphiphan et al. (2009)	0.000	0.424	0.124	0.645	0.141	0.224	0.516	0.183	0.149	0.258	0.183	0.135	0.000	0.183	0.405	0.000	0.135	[illegible]	[illegible]
Rico et al. (2013)	0.000	0.559	0.196	0.510	0.112	0.354	0.204	0.144	0.471	0.408	0.144	0.320	0.158	0.289	0.213	0.000	0.213	[illegible]	[illegible]
Ramazani and Vahdat-Nejad (2014)	0.289	0.316	0.139	0.289	0.158	0.250	0.289	0.000	0.167	0.289	0.408	0.151	0.224	0.204	0.302	0.204	0.302	[illegible]	[illegible]
Nassar et al. (2016)	0.000	0.141	0.372	0.258	0.424	0.224	0.258	0.000	0.000	0.000	0.000	0.135	0.000	0.000	0.000	0.183	0.270	[illegible]	[illegible]
Rauscher et al. (2009)	0.385	0.316	0.462	0.192	0.316	0.000	0.000	0.136	0.333	0.192	0.272	0.402	0.447	0.272	0.000	0.408	0.302	[illegible]	[illegible]
Bifulco et al. (2014) - Singapore	0.258	0.141	0.248	0.129	0.141	0.224	0.000	0.183	0.149	0.000	0.183	0.135	0.000	0.183	0.000	0.183	0.135	[illegible]	[illegible]
Bifulco et al. (2014) - Amsterdam	0.000	0.183	0.480	0.167	0.183	0.289	0.000	0.000	0.000	0.000	0.000	0.348	0.000	0.000	0.000	0.236	0.000	[illegible]	[illegible]
Barba et al. (2013)	0.236	0.387	0.226	0.236	0.129	0.204	0.236	0.333	0.544	0.471	0.667	0.369	0.548	0.333	0.246	0.167	0.246	[illegible]	[illegible]
Arkian et al. (2014)	0.204	0.335	0.196	0.204	0.224	0.177	0.204	0.433	0.471	0.408	0.433	0.213	0.632	0.289	0.213	0.144	0.213	[illegible]	[illegible]
Santa and Gómez-Skarmeta (2009)	0.000	0.112	0.196	0.102	0.000	0.000	0.000	0.144	0.354	0.204	0.289	0.107	0.474	0.577	0.107	0.000	0.426	[illegible]	[illegible]
Panagiotopoulos and Dimitrakopoulos (2019)	0.192	0.211	0.277	0.289	0.105	0.167	0.385	0.136	0.111	0.192	0.000	0.101	0.000	0.136	0.201	0.000	0.201	[illegible]	[illegible]
Figueiredo et al. (2001)	0.000	0.424	0.124	0.516	0.141	0.224	0.516	0.183	0.149	0.258	0.000	0.135	0.000	0.000	0.405	0.000	0.135	[illegible]	[illegible]
Atasoy et al. (2015)	0.000	0.632	0.092	0.481	0.211	0.167	0.000	0.272	0.333	0.192	0.272	0.503	0.298	0.272	0.201	0.272	0.302	[illegible]	[illegible]
Hu et al. (2017)	0.204	0.447	0.294	0.510	0.447	0.530	0.204	0.289	0.471	0.204	0.433	0.533	0.316	0.289	0.426	0.433	0.320	[illegible]	[illegible]
Kolbe et al. (2017)	0.000	0.258	0.340	0.354	0.129	0.204	0.000	0.000	0.408	0.236	0.167	0.246	0.183	0.333	0.246	0.333	0.246	[illegible]	[illegible]
Parodi et al. (2016)	0.000	0.516	0.113	0.589	0.000	0.204	0.236	0.000	0.408	0.471	0.333	0.123	0.183	0.500	0.246	0.000	0.246	[illegible]	[illegible]
Subramanyam and Ashwath Kumar (2016)	0.000	0.516	0.226	0.589	0.387	0.408	0.000	0.167	0.272	0.000	0.167	0.369	0.183	0.333	0.123	0.333	0.246	[illegible]	[illegible]
Nakamura et al. (2014)	0.000	0.478	0.000	0.436	0.000	0.000	0.000	0.154	0.378	0.000	0.154	0.228	0.000	0.000	0.342	0.000	0.114	[illegible]	[illegible]
Narayanan et al. (2014)	0.000	0.141	0.124	0.129	0.141	0.000	0.000	0.000	0.149	0.258	0.183	0.135	0.200	0.183	0.135	0.000	0.539	[illegible]	[illegible]
Wang et al. (2014)	0.000	0.335	0.196	0.306	0.335	0.354	0.204	0.433	0.354	0.204	0.000	0.213	0.158	0.289	0.213	0.577	0.107	[illegible]	[illegible]
David et al. (2013) - Loading Zone	0.000	0.474	0.139	0.433	0.000	0.250	0.289	0.204	0.333	0.289	0.000	0.151	0.000	0.204	0.302	0.000	0.000	[illegible]	[illegible]
David et al. (2013) - Communicating Bus Stop	0.000	0.158	0.000	0.289	0.000	0.000	0.000	0.000	0.333	0.289	0.204	0.452	0.224	0.204	0.151	0.000	0.302	[illegible]	[illegible]
Parundekar and Oguchi (2012)	0.000	0.200	0.000	0.456	0.000	0.000	0.000	0.129	0.316	0.183	0.258	0.191	0.141	0.258	0.381	0.000	0.572	[illegible]	[illegible]
Werther and Hoch (2012)	0.309	0.507	0.222	0.540	0.254	0.134	0.154	0.218	0.445	0.154	0.109	0.242	0.359	0.109	0.242	0.218	0.403	[illegible]	[illegible]
Saha and Chaki (2011)	0.204	0.112	0.196	0.306	0.000	0.000	0.000	0.144	0.354	0.204	0.144	0.107	0.158	0.289	0.107	0.144	0.320	[illegible]	[illegible]
Sadoun and Al-Bayari (2007)	0.000	0.258	0.226	0.354	0.258	0.204	0.236	0.167	0.136	0.236	0.167	0.246	0.183	0.167	0.000	0.333	0.369	[illegible]	[illegible]
Gena and Torre (2004)	0.480	0.263	0.385	0.240	0.175	0.000	0.160	0.226	0.370	0.160	0.226	0.334	0.248	0.226	0.084	0.226	0.251	[illegible]	[illegible]
Goto and Kambayashi (2002)	0.000	0.224	0.000	0.510	0.000	0.000	0.000	0.144	0.236	0.000	0.144	0.107	0.000	0.144	0.426	0.000	0.533	[illegible]	[illegible]

	Ramesh et al. (2013)	Al-Sultan et al. (2013)	Zarza et al. (2013)	Wang et al. (2013)	Fuchs et al. (2008)	Woernd and Eigner (2007)	Alghamdi (2012)	Hoogendoorn et al. (2012)	Ngai et al. (2012)	Baltrunas et al. (2011)	Raphiphan et al. (2009)	Rico et al. (2013)	Ramazani and Vahdat-Nejad (2014)	Nassar et al. (2016)	Rauscher et al. (2009)	Bifulco et al. (2014) - Singapore	Bifulco et al. (2014) - Amsterdam	Barba et al. (2013)	Arkian et al. (2014)
da Silva et al. (2018)	0.218	0.365	0.289	0.192	0.408	0.183	0.289	0.577	0.000	0.192	0.000	0.000	0.289	0.000	0.385	[illegible]	[illegible]	[illegible]	[illegible]
Vieira et al. (2011)	0.359	0.500	0.316	0.316	0.112	0.300	0.316	0.316	0.456	0.316	0.424	0.559	0.316	0.141	0.316	[illegible]	[illegible]	[illegible]	[illegible]
Kannan et al. (2010)	0.419	0.526	0.277	0.370	0.588	0.263	0.277	0.277	0.240	0.277	0.124	0.196	0.139	0.372	0.462	[illegible]	[illegible]	[illegible]	[illegible]
Meier et al. (2006)	0.327	0.365	0.144	0.289	0.000	0.365	0.144	0.144	0.417	0.385	0.645	0.510	0.289	0.258	0.192	[illegible]	[illegible]	[illegible]	[illegible]
Aguirre et al. (2017)	0.478	0.200	0.158	0.316	0.224	0.200	0.158	0.316	0.274	0.105	0.141	0.112	0.158	0.424	0.316	[illegible]	[illegible]	[illegible]	[illegible]
Younes et al. (2016)	0.000	0.158	0.000	0.167	0.177	0.000	0.000	0.000	0.144	0.167	0.224	0.354	0.250	0.224	0.000	[illegible]	[illegible]	[illegible]	[illegible]
Sukode and Gite (2015)	0.218	0.365	0.000	0.385	0.000	0.000	0.000	0.000	0.000	0.385	0.516	0.204	0.289	0.258	0.000	[illegible]	[illegible]	[illegible]	[illegible]
Sujitha and Punitha (2014)	0.309	0.258	0.204	0.136	0.289	0.000	0.204	0.000	0.118	0.272	0.183	0.144	0.000	0.000	0.136	[illegible]	[illegible]	[illegible]	[illegible]
Zardosht et al. (2014)	0.504	0.527	0.500	0.222	0.236	0.422	0.500	0.333	0.289	0.222	0.149	0.471	0.167	0.000	0.333	[illegible]	[illegible]	[illegible]	[illegible]
Khekare and Sakhare (2012)	0.218	0.365	0.289	0.192	0.000	0.183	0.289	0.289	0.167	0.385	0.258	0.408	0.289	0.000	0.192	[illegible]	[illegible]	[illegible]	[illegible]
Maslekar et al. (2011)	0.463	0.516	0.612	0.136	0.289	0.258	0.612	0.408	0.118	0.136	0.183	0.144	0.408	0.000	0.272	[illegible]	[illegible]	[illegible]	[illegible]
Fogue et al. (2011)	0.342	0.286	0.302	0.101	0.213	0.286	0.302	0.302	0.261	0.302	0.135	0.320	0.151	0.135	0.402	[illegible]	[illegible]	[illegible]	[illegible]
Ghaffarian et al. (2012)	0.507	0.283	0.671	0.149	0.158	0.283	0.447	0.447	0.258	0.149	0.000	0.158	0.224	0.000	0.447	[illegible]	[illegible]	[illegible]	[illegible]
Elhadef (2015)	0.463	0.387	0.408	0.000	0.289	0.129	0.408	0.204	0.236	0.136	0.183	0.289	0.204	0.000	0.272	[illegible]	[illegible]	[illegible]	[illegible]
Alazawi et al. (2011)	0.000	0.286	0.000	0.101	0.000	0.191	0.000	0.000	0.000	0.201	0.405	0.213	0.302	0.000	0.000	[illegible]	[illegible]	[illegible]	[illegible]
Bergan et al. (1998)	0.309	0.129	0.204	0.408	0.433	0.129	0.408	0.204	0.354	0.136	0.000	0.000	0.204	0.183	0.408	[illegible]	[illegible]	[illegible]	[illegible]
Alhammad et al. (2012)	0.342	0.191	0.302	0.101	0.107	0.477	0.302	0.151	0.435	0.201	0.135	0.213	0.302	0.270	0.302	[illegible]	[illegible]	[illegible]	[illegible]
Bae and Olariu (2010)	0.655	0.548	0.577	0.385	0.408	0.183	0.577	0.289	0.000	0.192	0.258	0.000	0.289	0.258	0.192	[illegible]	[illegible]	[illegible]	[illegible]
Alghamdi et al. (2012)	0.571	0.478	0.378	0.504	0.267	0.359	0.378	0.378	0.218	0.252	0.338	0.267	0.189	0.338	0.252	[illegible]	[illegible]	[illegible]	[illegible]
Ramesh et al. (2013)	1.000	0.478	0.567	0.378	0.267	0.359	0.567	0.567	0.327	0.252	0.169	0.134	0.189	0.338	0.630	[illegible]	[illegible]	[illegible]	[illegible]
Al-Sultan et al. (2013)	0.478	1.000	0.474	0.316	0.447	0.200	0.474	0.316	0.091	0.632	0.424	0.447	0.316	0.141	0.316	[illegible]	[illegible]	[illegible]	[illegible]
Zarza et al. (2013)	0.567	0.474	1.000	0.167	0.354	0.316	0.750	0.500	0.144	0.167	0.000	0.177	0.250	0.000	0.333	[illegible]	[illegible]	[illegible]	[illegible]
Wang et al. (2013)	0.378	0.316	0.167	1.000	0.354	0.316	0.167	0.333	0.385	0.333	0.298	0.236	0.500	0.298	0.333	[illegible]	[illegible]	[illegible]	[illegible]
Fuchs et al. (2008)	0.267	0.447	0.354	0.354	1.000	0.112	0.354	0.177	0.102	0.354	0.000	0.125	0.177	0.000	0.354	[illegible]	[illegible]	[illegible]	[illegible]
Woernd and Eigner (2007)	0.359	0.200	0.316	0.316	0.112	1.000	0.316	0.474	0.365	0.211	0.000	0.335	0.158	0.283	0.527	[illegible]	[illegible]	[illegible]	[illegible]
Alghamdi (2012)	0.567	0.474	0.750	0.167	0.354	0.316	1.000	0.500	0.144	0.167	0.000	0.177	0.250	0.000	0.333	[illegible]	[illegible]	[illegible]	[illegible]
Hoogendoorn et al. (2012)	0.567	0.316	0.500	0.333	0.177	0.474	0.500	1.000	0.289	0.167	0.000	0.177	0.250	0.224	0.500	[illegible]	[illegible]	[illegible]	[illegible]
Ngai et al. (2012)	0.327	0.091	0.144	0.385	0.102	0.365	0.144	0.289	1.000	0.192	0.129	0.408	0.144	0.516	0.481	[illegible]	[illegible]	[illegible]	[illegible]
Baltrunas et al. (2011)	0.252	0.632	0.167	0.333	0.354	0.211	0.167	0.167	0.192	1.000	0.447	0.471	0.167	0.298	0.333	[illegible]	[illegible]	[illegible]	[illegible]
Raphiphan et al. (2009)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Rico et al. (2013)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Ramazani and Vahdat-Nejad (2014)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Nassar et al. (2016)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Rauscher et al. (2009)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Bifulco et al. (2014) - Singapore	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Bifulco et al. (2014) - Amsterdam	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Barba et al. (2013)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Arkian et al. (2014)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Santa and Gómez-Skarmeta (2009)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Panagiotopoulos and Dimitrakopoulos (2019)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Figueiredo et al. (2001)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Atasoy et al. (2015)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Hu et al. (2017)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Kolbe et al. (2017)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Parodi et al. (2016)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Subramanyam and Ashwath Kumar (2016)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Nakamura et al. (2014)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Narayanan et al. (2014)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Wang et al. (2014)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
David et al. (2013) - Loading Zone	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
David et al. (2013) - Communicating Bus Stop	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Parundekar and Oguchi (2012)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Werther and Hoch (2012)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Saha and Chaki (2011)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Sadoun and Al-Bayari (2007)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Gena and Torre (2004)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Goto and Kambayashi (2002)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

	Santa and Gómez-Skarmeta (2009)	Panagiotopoulos and Dimitrakopoulos (2019)	Figueiredo et al. (2001)	Atasoy et al. (2015)	Hu et al. (2017)	Kolbe et al. (2017)	Parodi et al. (2016)	Subramanyam and Ashwath Kumar (2016)	Nakamura et al. (2014)	Narayanan et al. (2014)	Wang et al. (2014)	David et al. (2013) - Loading Zone	David et al. (2013) - Communicating Bus Stop	Parundekar and Oguchi (2012)	Werther and Hoch (2012)	Saha and Chaki (2011)	Sadoun and Al-Bayari (2007)	Gena and Torre (2004)	Goto and Kambayashi (2002)
da Silva et al. (2018)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Vieira et al. (2011)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Kannan et al. (2010)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Meier et al. (2006)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Aguirre et al. (2017)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Younes et al. (2016)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Sukode and Gite (2015)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Sujitha and Punitha (2014)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Zardosht et al. (2014)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Khekare and Sakhare (2012)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Maslekar et al. (2011)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Fogue et al. (2011)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Ghaffarian et al. (2012)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Elhadef (2015)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Alazawi et al. (2011)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Bergan et al. (1998)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Alhammad et al. (2012)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Bae and Olariu (2010)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Alghamdi et al. (2012)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Ramesh et al. (2013)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Al-Sultan et al. (2013)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Zarza et al. (2013)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Wang et al. (2013)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Fuchs et al. (2008)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Woernd and Eigner (2007)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Alghamdi (2012)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Hoogendoorn et al. (2012)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Ngai et al. (2012)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Baltrunas et al. (2011)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Raphiphan et al. (2009)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Rico et al. (2013)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Ramazani and Vahdat-Nejad (2014)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Nassar et al. (2016)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Rauscher et al. (2009)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Bifulco et al. (2014) - Singapore	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Bifulco et al. (2014) - Amsterdam	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Barba et al. (2013)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Arkian et al. (2014)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Santa and Gómez-Skarmeta (2009)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Panagiotopoulos and Dimitrakopoulos (2019)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Figueiredo et al. (2001)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Atasoy et al. (2015)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Hu et al. (2017)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Kolbe et al. (2017)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Parodi et al. (2016)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Subramanyam and Ashwath Kumar (2016)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Nakamura et al. (2014)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Narayanan et al. (2014)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Wang et al. (2014)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
David et al. (2013) - Loading Zone	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
David et al. (2013) - Communicating Bus Stop	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Parundekar and Oguchi (2012)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Werther and Hoch (2012)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Saha and Chaki (2011)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Sadoun and Al-Bayari (2007)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Gena and Torre (2004)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Goto and Kambayashi (2002)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Source: The author (2022)

APPENDIX C – EXAMPLES OF CONTEXT ELEMENTS

The items in bold are examples of context elements in each category of the taxonomy.

- Time
  - Local Time
    - **Local Time/Date**
    - **Coordinated Time/Date**
    - **Day of the Week**
    - **Timestamp**
  - Schedule
    - **Personal Schedule**
    - **Delivery Tables**
    - **Holiday List**
    - **Bus lines timetable**
  - Travel time
    - **Time elapsed since travel start**
    - **Time elapsed since last rest stop**
    - **Estimated Time to Arrival**
- Location
  - Geographic Coordinate
    - **Latitude**
    - **Longitude**
    - **Altitude**
  - Symbolic Coordinate
    - **Cellular Network Base Station Id**
    - **Wifi SSID**
    - **Beacon Id**
  - Geographic (Semantic) Location

    * **Address**
    * **Road Name**
    * **Floor (i.e.: in multi-story car parks)**
    * **Train station Name / Id**
    * **Bus stop Id**
    * **Traffic Light Id**
  - Location Type
    * **Kind of Place**
    * **Place Service**
  - Route
    * **Start and End point**
    * **Intermediate stops**
    * **Bus line itinerary**
  - Distance
    * **Total distance traveled**
    * **Distance traveled in current journey**
    * **Distance traveled since last rest stop**
    * **Distance to destination**
    * **Distance to next waypoint**
- Identity
  - Vehicle Id
    * **VIN**
    * **License Plate**
    * **National Registration Number**
    * **Brand**
    * **Model**
    * **Year**
    * **Electronic tag Id**

- Vehicle Dimensions
  * **Length**
  * **Width**
  * **Height**
  * **Empty Weight**
  * **Max Weight**
  * **Wheelbase**
- Vehicle Type
  * **Vehicle Category**
  * **EuroCar Segment Classification**
  * **US EPA Classification**
- Vehicle Mechanical Attributes
  * **Engine Displacement**
  * **Suspension Type**
  * **Gear Characteristics**
  * **Axels**
  * **Traction**
- Person Id
  * **Possession-based (Official Document, Ticket, Card SSN)**
  * **Knowledge-based (User/password)**
  * **Biometric-based (Fingerprint, face, iris**
- Passenger Id
  * **Person Id***
- Driver Id
  * **Person Id***
- Driver Experiencee
  * **Is in training?**
  * **Is recently licensed?**
  * **Years Licensed**

    * **Vehicle Categories allowed to drive**
    * **Special training (working driver, dangerous goods)**
    * **Safety history**
- Profile
    * **Name**
    * **Age**
    * **Birthday**
    * **Profession**
    * **Gender**
    * **Infotainment preferences (Music style, Radio stations, cellphone answering policy)**
    * **General preferences (Route type, route distance vs, time, Toll policy, Gas stations, shops)**
    * **Addresses**
    * **Email**
    * **Phone number**
    * **Social Media Addresses**
    * **Allergies**
- Physical Attributes
    * **Height**
    * **Weight**
    * **Physical abilities / Special needs**
- Place Id
    * **Geographic Coordinate***
    * **Symbolic Coordinate***
    * **Geographic (Semantic) Location***
- Place Attributes
    * **Length**
    * **Width**

    * **Height**
    * **Max allowed weight**
    * **Allowed vehicle types**
    * **Minimum age required**
    * **Opening Hours**
    * **Other restrictions**
  - Road Characteristics
    * **Pavement type**
    * **Number of lanes**
    * **Default maximum speed**
  - Payload Dimensions
    * **Length**
    * **Width**
    * **Height**
    * **Weight**
    * **Has special format / needs special handling?**
  - Payload Type
    * **Is live?**
    * **Is perishable?**
    * **Is fragile?**
    * **Is Dangerous?**
    * **Hazardous Materials Classification**
  - Payload Id
    * **Barcode / QR-Code**
    * **Tracking IDs**
  - Device Type
    * **Is mobile?**
    * **Is inside/part of the vehicule?**
    * **Is network device?**

    * **Is traffic infrastructure?**
    * **Power supply type**
    * **Power**
  - Device UID
    * **MAC address**
    * **Sensor Id**
    * **Traffic Light Id**

- Activity
  - Driver Status
    * **Pulse Rate**
    * **Temperature**
    * **Blood alcohol content**
    * **Tiredness**
    * **Mood**
  - Driving Task
    * **Hands position**
    * **Eyes direction**
    * **Throttle pressure**
    * **Break pressure**
    * **Current gear**
  - Driving External Task
    * **Is talking to other passengers?**
    * **Is using mobile?**
    * **Is talking to pedestrian?**
    * **Is interacting with drive-thru/tool booth?**
  - Diver Infotainment Task
    * **Is radio on?**
    * **Current radio volume**

    * **Is changing volume?**
    * **Is changing mode?**
    * **Is changing station**
    * **GPS has route set?**
    * **Is handling GPS?**
  - Vehicle Activity
    * **Is in autonomous mode?**
    * **Is parking?**
    * **Is stopped?**
    * **Is in service? (taxi ride / transporting passengers, cargo or patients / responding to an emergency)**
    * **Is in emergency?**
    * **Ambient noise level**
    * **Internal temperature**
  - Mechanical Status
    * **Fuel tank level**
    * **Current gear**
    * **RPM**
    * **Engine load**
    * **Oil temperature**
    * **Coolant Temperature**
    * **Throttle position**
    * **Fuel type/mix**
    * **Lights status**
    * **Blinkers status**
    * **Failure/malfunction error code**
    * **Has crashed?**
    * **Airbag deployment status**
    * **Seatbelt status**
    * **Any OBD-II readable information**

- Vehicle Movement
  - **Speed**
  - **Average speed**
  - **Angular velocity**
  - **Acceleration**
  - **Direction**
  - **Has rolled over?**
  - **Had sudden accelaration/decelleration?**
- Passenger
  - **Current activity**
  - **Objectives**
  - **Seat Occupied**
  - **Is sat?**
  - **Is using seatbelt?**
  - **Is wake?**
  - **Pulse rate**
  - **Temperature**
  - **Mood**
- Traffic Device
  - **Traffic light status**
  - **Traffic light timer**
  - **Traffic sign type**
  - **Message board message**
  - **Toll plaza status**
  - **Barrier status**
- Traffic Condition
  - **Traffic intensity**
  - **Accident notifications**
  - **Abnormal conditions (car stopped in the road, potholes)**

- Road Availability
  * **Road works**
  * **Road blocks**
  * **Vehicle types allowed in the road**
- Cargo
  * **Temperature**
  * **Movement**
- Pedestrian Movement
  * **Acceleration**
  * **Direction**
  * **Speed**
- Pedestrian Role
  * **Is waiting?**
  * **Is future passenger?**
- Weather
  * **Temperature**
  * **Rain/Snow status**
  * **Moisture**
  * **Wind speed**
  * **Tide****
- Lanes
  * **Lane Status (Open/Closed)**
  * **Lane direction (for reversible lanes)**
  * **Vehicle types allowed in lane**
- Point of Interest
  * **Surrounding POIs**
  * **Services offered**
  * **Service values**
  * **Is open?**

- Infrastructure
  - **Surrounding Infrastructure Devices**
  - **Traffic Device***
  - **Road Side Unit***
- Traffic Controller
  - **Surrounding Traffic Controllers**
  - **Actions**
  - **Commands**
- Law Enforcement
  - **Surrounding Law Enforcement Agents**
  - **Commands**
  - **Type**
  - **Is human or machine?**
- Surrounding Cyclist
  - **Surrounding Cyclists**
  - **Pedestrian Movement***
- Surrounding Driver
  - **Surrounding Drivers**
  - **Profile***
- Surrounding Passenger
  - **Surrounding Passengers**
  - **Location***
  - **Travel Time***
  - **Route***
  - **Profile***
- Surrounding Pedestrian
  - **Surrounding Pedestrians**
  - **Pedestrian Movement***
  - **Profile***

- Surrounding Vehicles
  - **Surrounding Vehicles**
  - **Vehicle Movement***
  - **Mechanical Status***
  - **Vehicle Activity***
  - **Location***
  - **Route***
- Social Media
  - **Events nearby**
  - **Traffic events**
  - **Persons nearby**
  - **Businesses nearby**
- Network Connectivity
  - **Bandwidth**
  - **Network type**
  - **Connectivity level (no access, local, internet)**
- Network Status
  - **Latency**
  - **Packet loss**
  - **Jitter**
- Road Side Unit
  - **Connectivity***
  - **Location***
- Peer Device
  - ***Any CE abount a network peer device***

* Any CE from the category (or its subcategories) is also a CE for this category

** While not a weather condition, we consider that it is fit to put Tide in this category, since applications which might use Tide information are probably using other weather information too.

APPENDIX D – KNOWLEDGE BASE 1.1

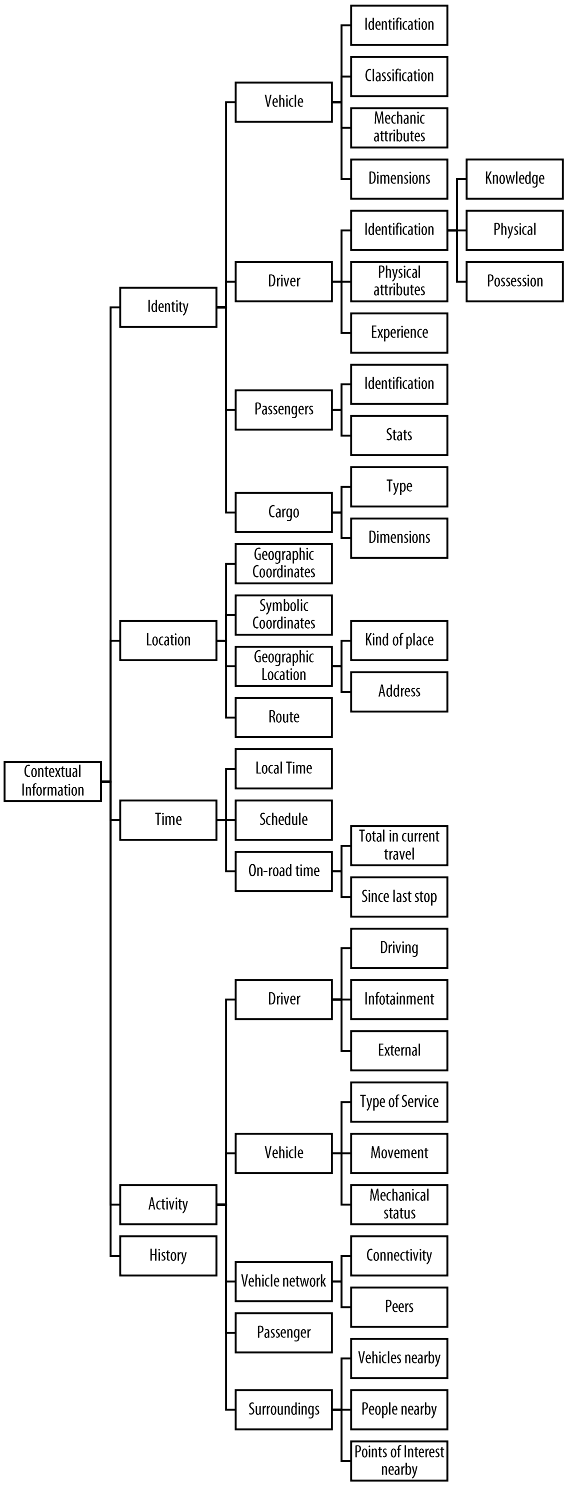
Frame 9 – Knowledge base of Projects x Context Element Categories - Version 1.1

Moovit (SANTOS; NIKOLAEV, 2021)	Uber	Waze	Sabet et al. (2020)	Haque et al. (2018)	Ali, Muhammad e Khan (2020)	Özkul, Capuni e Domnori (2018)	Chavhan et al. (2020)	Tao et al. (2020)	Jiang et al. (2020)	Chavhan et al. (2021)	Mondal e Rehena (2021)	Goto e Kambayashi (2002)	Gena e Torre (2004)	Sadoun e Al-Bayari (2007)	Saha e Chaki (2011)	Werther e Hoch (2012)	Parundekar e Oguchi (2012)	David et al. (2013) - Bus stop	David et al. (2013) - Loading zone	Wang, David e Chalon (2014)	Narayanan et al. (2014)	Nakamura et al. (2014)	Hu et al. (2015)	Subramanyam e Kumar (2016)	Parodi et al. (2016)	Kolbe et al. (2017)	Hu et al. (2017)	Atasoy et al. (2015)	Figueiredo et al. (2001)	Panagiotopoulos e Dimitrakopoulos (2019)	Santa e Gómez-Skarmeta (2009)	Arkian, Atani e Kamali (2014)	Barba et al. (2013)	Bifulco, Amitrano e Tregua (2014) - Amsterdam	Bifulco, Amitrano e Tregua (2014) - Singapore	Zhang, Cheng e Lin (2012)	Rauscher et al. (2009)	Nassar, Kamel e Karray (2016)	Ramazani e Vahdat-Nejad (2014)	Rico et al. (2013)	Raphiphan et al. (2009)	Baltrunas et al. (2011)	Ngai et al. (2012)	Hoogendoorn, Breukink e Arem (2012)	Alghamdi (2012)	Woernd e Eigner (2007)	Fuchs, Rass e Kyamakya (2008)	Wang, Jiang e Mu (2013)	Zarza et al. (2013)	Al-Sultan, Al-Bayatti e Zedan (2013)	Ramesh, Vidya e Pradeep (2013)	Alghamdi, Shakshuki e Sheltami (2012)	Bae e Olariu (2010)	Alhammad, Siewe e Al-Bayatti (2012)	Bergan, Bushman e Taylor (1998)	Alazawi et al. (2011)	Elhadef (2015)	Ghaffarian, Fathy e Soryani (2012)	Fogue et al. (2011)	Maslekar et al. (2011)	Khekare e Sakhare (2012)	Zardosht, Beauchemin e Bauer (2014)	Sujitha e Punitha (2014)	Chen e Lu (2015)	Sukode e Gite (2015)	Younes, Boukerche e Mammeri (2016)	Aguirre et al. (2017)	Meier, Harrington e Cahill (2006)	Johnson e Trivedi (2011)	Kannan, Thangavelu e Kalivaradhan (2010)	Vieira, Caldas e Salgado (2011)	Silva, Borges e Vieira (2018)		
0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	Driver \ Status	Activity
0	1	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	Driver \ Driving Task	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Driver \ External Task	
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Driver \ Infotainment task	
1	1	1	0	0	0	0	0	0	1	1	0	0	1	0	1	0	0	0	1	1	0	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	Infrastructure \ Status \ Road Availability	
0	0	1	0	0	0	0	1	0	0	1	1	0	0	0	0	1	0	0	1	1	0	0	1	0	1	0	0	0	1	1	0	1	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	1	0	1	0	0	1	0	Infrastructure \ Status \ Traffic Condition	
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Passenger	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Cargo	
0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Pedestrian \ Movement	
0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	Pedestrian Role	
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Surroundings \ Point of Interest	
0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Social Media	
0	0	1	1	0	0	0	1	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0	Surroundings \ Weather	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Infrastructure \ Road Side Unit	
0	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	Surroundings \ Lanes	
0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	1	0	0	0	1	1	0	0	0	0	0	0	Infrastructure \ Traffic Device	
0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Surroundings \ Infrastructure	
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	Surroundings \ Traffic Participants \ Cyclist	
0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Surroundings \ Traffic Participants \ Driver	
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	Surroundings \ Traffic Participants \ Passen	
0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	Surroundings \ Traffic Participants \ Pedestr	
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Surroundings \ Traffic Controller	
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Surroundings \ Law Enforcement	
0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Surroundings \ Point of Interest	
1	0	1	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	1	1	1	1	0	0	1	0	0	1	0	1	1	0	0	0	0	0	0	1	0	0	Surroundings \ Traffic Participants \ Vehicle	
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	Network \ Status	
0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Network \ Connectivity	
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	Network \ Peer Device	
0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	1	0	0	Vehicle \ Activity	
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1	0	1	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	Vehicle \ Mechanical Status	
0	1	1	0	0	1	1	1	0	1	1	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1	1	0	0	0	1	0	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	1	0	0	1	1	1	0	1	0	0	0	0	1	0	1	1	1	1	Vehicle \ Movement	
0	0	1	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	Place \ Road Characteristics	Identity
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Device \ Device Type	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	Device \ Device UID	
0	1	1	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	1	0	1	1	1	0	0	1	1	0	0	0	1	1	0	0	1	0	Vehicle \ Vehicle Id	
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	Vehicle \ Dimensions	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	Vehicle \ Mechanical Attributes	
0	1	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	1	1	1	0	0	0	0	0	1	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	1	1	1	0	1	1	0	Vehicle \ Classification	
1	0	1	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	Place \ Place Id	
1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Place \ Place Attributes	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Payload \ Payload Id	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Payload \ Payload Dimension	
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Payload \ Payload Type	
0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	Person \ Identification \ Passenger Id	
0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Person \ Identification \ Person Id	
0	1	0	0	0	0	0	1	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Person \ Identification \ Driver Id	
1	1	1	1	0	0	0	1	0	0	0	0	1	1	1	0	1	1	0	0	0	1	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	1	0	1	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	Person \ Information \ Profile	
0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Person \ Information \ Driver Experience	
0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	Person \ Information \ Physical Attributes	
1	1	1	0	0	0	1	1	0	0	1	1	0	1	1	1	0	1	1	0	0	1	0	1	0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	1	0	1	1	1	1	1	0	0	1	1	1	0	0	1	0	0	1	1	1	1	1	1	0	0	0	0	0	1	0	0	1	0	Coordinate \ Geographic	Location
1	0	1	0	1	1	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	Geographic Location	
0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	Location Type	
1	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	Path \ Route	
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	Coordinate \ Symbolic	
1	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	Path \ Distance	
1	1	1	1	1	0	0	1	0	1	0	0	1	1	0	0	1	1	0	1	0	0	1	1	0	0	0	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	1	0	1	0	0	0	1	0	0	1	0	0	1	1	0	0	0	0	1	0	0	1	0	Time	Time
1	1	1	1	1	0	0	1	0	0	0	1	1	0	0	0	1	0	0	1	0	0	0	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	Schedule	
1	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Travel Time	

Source: The author (2022)

APPENDIX E – TAXONOMY OF CONTEXT ELEMENTS FOR ITS V0.1

Figure 21 – Taxonomy of Context Elements for ITS - Version 0.1



Source: The author (2022)