In-Vehicle System for Adaptive Filtering of Notifications

Elena Bautu
Ovidius University
Mamaia Blvd. 124,
Constanta, Romania
ebautu@gmail.com

Carmen I. Tudose
Ovidius University
Mamaia Blvd. 124,
Constanta, Romania
tudosecarmenionela@gmail.com

Crenguta M. Puchianu
Ovidius University
Mamaia Blvd. 124,
Constanta, Romania
crenguta.bogdan@gmail.com

ABSTRACT
Nowadays, there is a growing interest for drivers and car passengers to have adaptive smartphone applications to the driving context of the car.
In this paper we present a software system that filters notifications launched by mobile phone depending on the driver's or co-driver's interests, needs or preferences.
The filtering is adaptive in the sense that it changes according to the driving conditions determined by the system using data received from the GPS system and accelerometer of smartphone.
The system has been developed using "separation of concerns" principle that guarantees an iterative and incremental development of system. Also, it has been tested until now by 28 users who appreciated the system's quality considering their answers on the usability test.

Author Keywords
In-vehicle system, mobile interaction, software engineering, design patterns.

ACM Classification Keywords
H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION
Nowadays, there is a growing need for drivers and car passengers to have adaptive smartphone applications to the driving context of the car.

The reason is that many apps (e.g., Gmail, SMS, Facebook, WhatsApp, etc.) are running on the phone, or the user is subscribed to different sites (e.g., YouTube, cooking sites, fashion or travel blogs) to receive notifications on their phone.

While driving, these notifications can be disruptive to the participant, especially in certain situations where the driver must be extremely careful such as accelerating or going at high speed (over 130 km/h), change direction in a curve, or he/she must suddenly brake, because there was an obstacle in his/her path.

In these situations, it would be desirable for the driver's/ co-driver's phone to emit as few calls or notifications, and the interaction with the phone to be as small as possible.

In this paper we present a software system that filters notifications launched by mobile phone depending on the driver's or co-driver's interests, needs, or preferences.

Before driving the car, the user set two kind of notifications and calls:

- those he/she would like to be announced under normal driving conditions,
- those he/she would like to be announced under conditions of high risk of driving.

Then the system filters the notifications that it changes according to the driving conditions determined by the system using data received from the GPS system and accelerometer of smartphone.

Another category of potential users would be people, who do not want to receive all sorts of notifications, but they are still waiting for an important phone from a certain person and therefore they cannot set the phone to silent mode.

STATE OF THE ART
The attractiveness of relying on personalized services when on board of a car is discussed, via an appropriate software application, in [7]. The paper proposes a tourism oriented app to be used in an in-vehicle scenario that provides „the "right" service at the "right" time and in the "right" way". Having as main goal a system that offers tourist information and distracts the driver in a minimal manner, the authors discuss the main design choices in order to achieve it.

Driver behavior is the subject of the survey presented in [23]. Various mathematical identification methods or modeling methods used for deriving driver behavior models, which are supposed to be further integrated into advanced driver assistance systems. The authors decompose the process of recognizing driver behavior in four stages, and exemplify with respect to driving specific tasks, such as steering or gear shifting.

The manner in which in-vehicle devices influence drivers' behavior in high-demand situations is studied in [24]. As a continuation, in [3] various auditory, vibro-tactile, and visual take-over requests are investigated, in the context of highly automated driving. Among the conclusions of this large crowdsourcing study, we mention: multimodal requests were favored in high-urgency scenarios; auditory take-over-requests were most preferred in low-urgency situations where, also, visual was preferred over vibration. Also in the context of autonomous driving, the authors of [10] discuss the selection of the human driver's longitudinal behavior at a non-signalized roundabout in an adaptive fashion, while taking into account the three objectives of safety, efficiency and comfort. They propose an algorithm that learns from human experience and then a multi-objective algorithm for decision making, used for selecting the optimal behavior in dynamic scenarios. The algorithmic approach is compared, with respect to performance, to human driving and the results are promising.

MOTIVATION

In addition to all the advantages of using mobile phones in society, their use also presents a number of disadvantages that can even lead to people's death when attention is directed to the phone and not to what is important.

Nowadays many driving accidents have happened because of the drivers distracted by some notifications received and displayed on their phones. According to a US traffic accident statistics [22], around 660,000 drivers are trying to use the mobile when they are at the wheel of the car during the day. Thus, the US National Security Council reports 1.6 million crashes each year due to the use of mobile phone while driving. In the United States, 1 in 4 car accidents are caused due to the sending of messages [22].

A serious accident occurred in Hungary on May 22, 2018 in which nine people lost their lives, blaming entirely on the bus driver focusing on the mobile phone and not on driving safely [16].

OUR SYSTEM

In order to foster the driver's attentiveness to the driving task, the ANOTIVE (Adaptive NOTifications In-VEhicle) system proposed in this paper allows the driver to manage information of interest, using a mobile phone as less as possible.

Development of software system

In the system development process, we based on the principle "separation of concerns", which in software engineering, is a mechanism for decomposing and structuring a system into modules, each of which manages a concern in a particular area of interest [15].

Concern-oriented software analysis

In order to identify the functional and nonfunctional requirements of the software system and to construct the functional model of our system, we firstly identified the software actors of our system and their concerns.

A software actor is a role "played" by one or more stakeholders (person, team, or organization) or even another software application in its interactions with our application. The role is is characterized by a set of properties and actions which each stakeholder in this role can exhibit or "plays" in the given context [5].

In the case of ANOTIVE application, we identified four software actors: driver, co-driver and the phone's GPS system and accelerometer.

Each human actor has one or more concerns regarding the usage of the system.

We used the concept of the stakeholder's concern as a problem-originated care of one or more stakeholders involved in the construction or evolution of an information system in its natural environment [5]. The care of a stakeholder may derive from his/her: 1) interest or responsibility in the construction or evolution of the software system in its environment, 2) wish to improve or modify something in the world for better matching his/her expectations, or 3) worrying about something wrong or undesired could occur.

Intuitively, we can define a problem as the tense, often conflictual, relationship between the things or the states of fact as perceived in the studied reality and the things and states imagined as they wish to be. In other words, we can see a problem as being constituted by the initial state of the studied reality and the final state we want to reach.

Following the suggestion of B. Moret [12], the initial state can be modeled as a finite set of parameters, which form the hypothesis of the problem. The final status is obtained if one or more questions are answered. These questions form the conclusion of the problem. We will call the problem specification the pair formed by the hypothesis and conclusion.

For example, the main concern of the driver during driving is the care to drive the car carefully to avoid an accident. From this concern we obtain other concerns that depend on the first one and existence of other resources which could obstruct the concern solving.

For example, if the driver's mobile phone is working and is placed in the car, other two concerns appear and depend on the first one:

- be as disturbed by notifications received from third party applications, such as Facebook, Gmail, WhatsApp, or any other notification automatically received by his mobile phone (e.g. automatic updates for installed software) (Table 1).

- not to lose notifications (calls, SMSs) from the people who the driver is interested in.

C2	*Name*: Care not to be disturbed by the phone during the driving activity	
Problem	*Hypothesis*: Driving started and mobile phone works	
	Conclusion: How can my mobile phone be set up to automatically prevent unwanted calls or notifications during the driving activity?	
Stakeholders: Driver, Co-driver		

Table 1. High level specification of the concern C2

C3	*Name*: Care to be quickly notified of important notifications during the driving activity	
Problem	*Hypothesis*: Driving started and mobile phone works	
	Conclusion: How can I set the phone to automatically announce me important calls or notifications during the driving activity?	
Stakeholders: Driver, Co-driver		

Table 2. High level specification of the concern C3

By defining concern as a care of one or more stakeholders in relation to a problem concerning the development or operation of a system, we do the following observations:

- being a concern, a concern is an intentional mental state of one or more stakeholders about a real-world problem;

- the problem has to be resolved and the associated concern will generally disappear.

Next, we will say that a concern is solved if the participant's care is met (satisfied), while also involving the problem of the concern. Solving any concern involves performing one or more actions, depending on the complexity of the problem contained, by one or more stakeholders. For this reason, we can see a concern as the source of one or more actions that will be undertaken by the stakeholders to solve the concern.

These actions of the stakeholders together with the actions of the system form the main flow of a software use case of the system. In software engineering, a software use case is a coherent unit of functionality provided by a classifier (system, subsystem, class) manifested through message sequences between the classifier and one or more software actors together with the actions made by the system [17].

In this way, each concern is solved by one or more software use cases. For instance, the software use case “Setting locked numbers” solves the concern C2 and is extended by other two use cases which solve the concerns of adding/removing of a phone number to/from the blocked list of numbers.

For instance, the main and alternative flows of the software use case “adding a phone number to the blocked list of numbers” is written in the Table 3.

User	System
1. Asks the system to set locked numbers	2. Allows numbers to be added to blocked numbers.
3. Asks for a new contact to be added to the locked list.	4. Allows the user to add a new phone number or to choose an existing number in the phonebook.
5. Choose a phonebook number [A1].	6. Save the number in blocked numbers list.
	7. Displays an operation completion message.
[A1] – Entering a new phone number 1. The system checks if the entered number is syntactically correct. 2. If so, the system displays an error message and the flow continues with step 2 of the main flow without saving the changes. 3. If the data is correct, then the flow continues with step 6 of the main flow.	

Table 3. The main and alternative flows of the software use case “adding a phone number to the blocked list of numbers”

These software use cases are graphically depict in the UML (Unified Modeling Language [14]) diagram use cases from the Figure 1. This diagram has been created using the Astah IDE [2].

We observe in Figure 1, the GPS system is a software actor because it is an external system of our application and interacts with it to send the coordinates of the current location of the phone and implicitly of the car.

This information is used by the ANOTIVE application to calculate the speed at which the machine moves. If the speed is above the 130km/h threshold or the driver has pressed the brake, the application automatically changes the filter notification options for the phone.

Another ANOTIVE application software actor also inside the smartphone is an accelerometer. If the phone is placed on the dashboard, the sensor sends information about that car is in a curve. Also in this scenario, the app automatically switches filtering options for phone notifications.

Software Design

The concern-oriented analysis has focused on learning to "do the right thing"; that is, identifying and understanding the functional and nonfunctional requirements of the ANOTIVE application. By contrast, the design work will stress " do the thing right" [17]; that is, skillfully designing a solution to satisfy the system requirements.

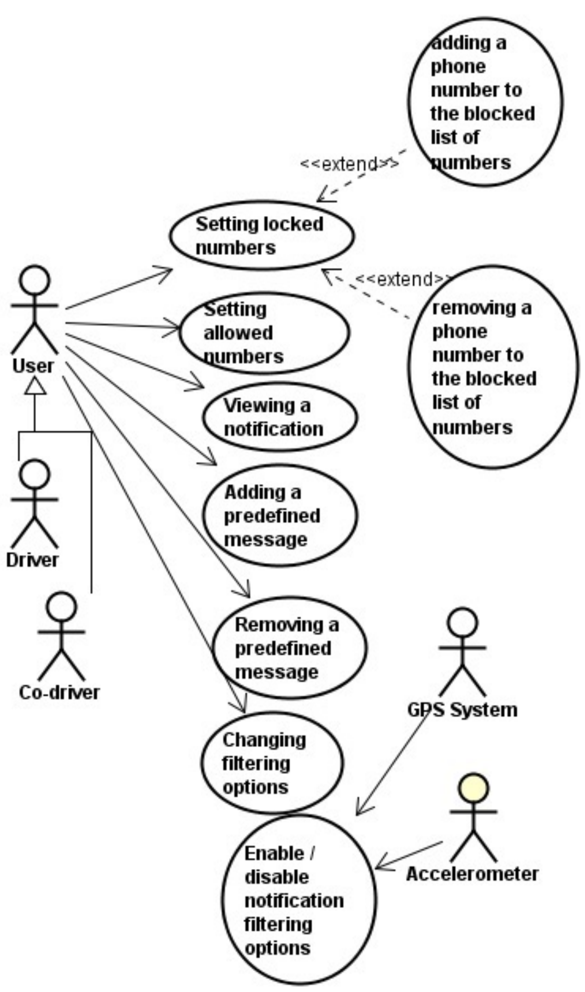
adding a phone number to the blocked list of numbers
<<extend>>
Setting locked numbers
<<extend>>
removing a phone number to the blocked list of numbers
User
Setting allowed numbers
Viewing a notification
Adding a predefined message
Driver
Removing a predefined message
Co-driver
GPS System
Changing filtering options
Enable / disable notification filtering options
Accelerometer

Figure 1. The UML software use cases diagram of the ANOTIVE system

The heart of this solution is the creation of the system software architecture, which contains from the static viewpoint the design classes and their relations. We constructed the software architecture of ANOTIVE system in an incrementally manner designing the software use cases by applying the Model-View-Controller (MVC) architectural pattern [8].

According to this pattern, the objects are classified in three categories: model, views, and controller objects. The classification criterion is given by the responsibilities of the objects from each category. View objects are objects with which user stakeholders interact directly, such as frames, forms, panels, and so on. The model objects eventually contain persistent information managed by the system. Many such objects come from the business objects of the domain model of the information system where the software system will operate. However, other objects from this category can also emerge during the design activity. Finally, the controller objects have the responsibility to manage the logical flow and the events produced by users in their interactions with the view objects. For instance, the part of the software architecture that contains the design classes used in.

In addition, in the construction process of the system's software architecture, we applied the GRASP design patterns, e.g. Expert, Creator, Low Coupling, High Cohesion, Pure Fabrication and Interface, and the creational pattern Singleton [9]. These design patterns provide a guarantee that we have obtained a quality architecture that applies basic design principles, such as low coupling or high cohesion of classes.

For instance, the UML diagram of classes obtained from the design of the software use case "adding a phone number to the blocked list of numbers" is presented in Figure 2.

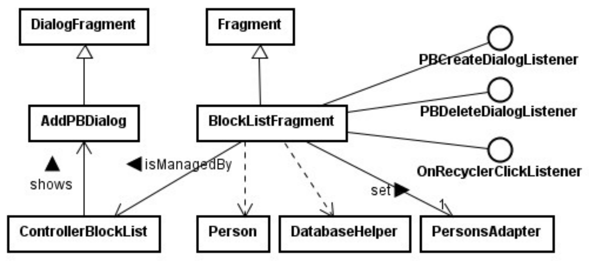

Figure 2. The UML class diagram obtained from the design of the software use case "adding a phone number to the blocked list of numbers"

Implementation

The classes obtained by the design of each software use case had been implemented in Java using the Android Studio development framework [1].

For instance, at the fourth step of the main flow (Table 3) of the software use case "adding a phone number to the blocked list of numbers", the application shows the screen from Figure 3.

The implementation in Android of the classes obtained in the design of the software use cases has been made with the following observations:

- view classes were implemented as fragments in which the controler classes were embedded;
- some of the model classes were implemented in Android activities, the other being implemented as common Java classes (e.g., Person, DatabaseHelper from Figure 2).

All the persistent data are stored in a SQLite database [21] that is installed on the user's smartphone when the application's apk file is installed.

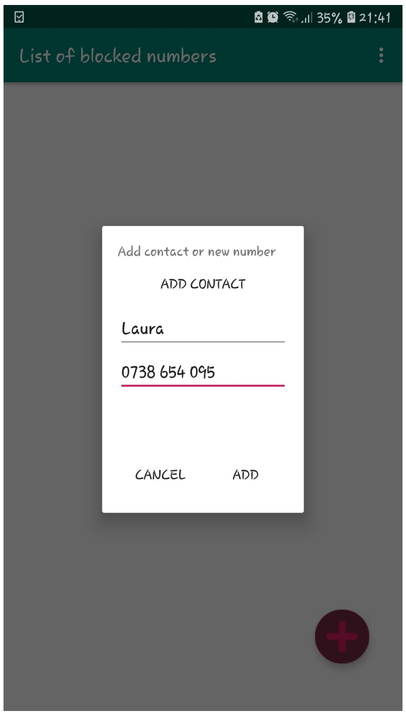

Figure 3. The screen corresponding the fourth step of the Table 3

Testing

Preliminary experimental evaluation of ANOTIVE was performed in a usability testing scenario that targeted users from two main categories:

- Active traffic participant, such as automobile drivers, as well as cyclists, roller blade skaters;
- Passive traffic participant, such as passengers.

Prior to filling the questionnaire, the application was showcased to the users and its functionalities were explained. Each participant downloaded a copy of the app on his/her personal mobile Android phone and used the app in a realistic manner, on their next trip. The only requirement was that the Android version installed on the device was greater that 26; the other various characteristics of the devices were not taken into consideration at this point.

The questionnaire was composed of a section gathering information about the profile of the users, a section regarding the usability of the app, and finally a section regarding the degree of utility and satisfaction perceived by the users [18]. Questions were designed on a Likert-type rating scale with 7 points [19], where users responded by specifying a level of agreement/disagreement with the statements comprised in the questionnaire. Activity related criteria, as well as performance criteria of the app were targeted by the second section [11]. The last two items are two open ended questions, meant to gather opinions from the users with respect to what they liked/disliked about the system.

The users were pooled from the students and personnel of the Faculty of Mathematics and Computer Science; hence they presumably have extended experience using mobile devices. At this point in the development of our in-vehicle system, we were interested in users that find no impediments in using modern technologies, such that their answers about the app are not biased by their own level of technology usage (all the users selected the answer 6 or 7 to the question “How often do you use mobile devices?”). The user sample consisted of 28 users – 13 female users and 15 men, of which 65% were younger than 30. The characteristics of the users sample are described in Figure 4. Among them, 17 users responded from the role of “active traffic participant” and 11 responded as “passive traffic participant”. The average score regarding mobile device use was 6.44. Future work on this project involves testing the app on a larger and more representative sample of users [11].

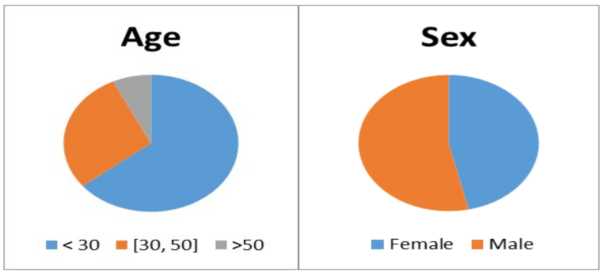

Figure 4. Characteristics of the users with respect to age and sex

The information displayed by the app was considered appropriate by the users (the mean score to the question is 5.92, with 86% of the users responding with 6 and 7). The application responded quite quickly to context updates (all scores were greater than 4).

The item Q9 of the questionnaire, “I think the use of the application did not distract me more than the phone / tablet would do in a normal use”, was agreed upon, with scores of 6 and 7, by 92.86% of the scores. 53.57% of users totally disagree with the statement that “I think interaction with the device requires too much physical effort.”, while 89.29% of users disagree (with scores 1 or 2) with the statement “I think the information displayed was confusing”.

Among the features appreciated with comments, we mention “The app blocks messages containing unwanted words, displays all blocked notifications in one place, I can choose to block notifications from any application I want” or “easy to use”. Things that were mentioned on the negative part: “The app performs with a lag”, “it is too simplistic”, and “it treats too few context conditions”.

The results of the evaluation survey are encouraging (see, for example, Figure 5 and Figure 6 for the summary of scores for two questions from the questionnaire). However, considering the characteristics of the user sample, with respect to age and level of experience with mobile apps, the sample cannot be considered as representative for the final target group of our system.

Further work will be devoted to quantifying the relation between user characteristics and the answers on the usability test, as well as performing repeated usability tests on a larger and more varied user sample.

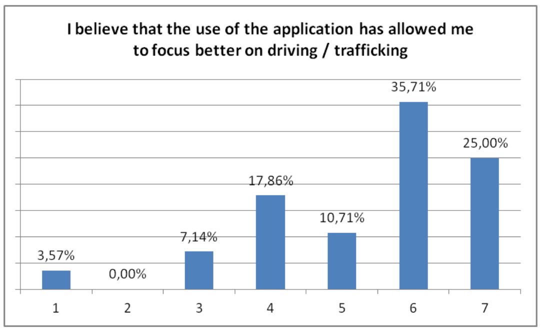

Figure 5. Summary of answers to item Q6. Average score 5.39, with a standard deviation of 1.52

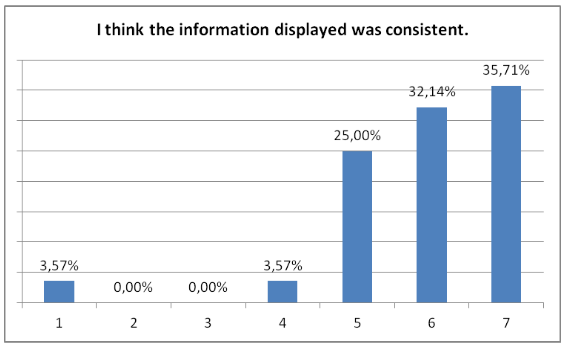

Figure 6. Summary of answers to item Q7. Average score 5.85, with a standard deviation of 1.29

RELATED WORK

For the development of the ANOTIVE application, we analyzed the applications in the same field, looking at the weaknesses of other applications, their way of using and functioning, but also what we could bring as an innovative element to existing applications. In the following are some existing applications in the same area with our system.

The "Notification Blocker & Cleaner & Heads-up Off" app primarily aims to delete notifications from the top bar of the phone screen. "Unnecessary notifications will be blocked and gathered in one place" [13] so that the top bar of the screen is kept free and the notification history can be viewed on a separate screen.

The "Calls Blacklist - Blocați apeluri și SMS-uri" application focuses on some types of notifications, such as calls or SMS messages. The main functionality is to block different calls according to user preferences, but also messages [6].

"Blocare apeluri" is an application for blocking unwanted calls. The user of this application will need to add a list of numbers to the "Black List", "White List" or to choose a blocking option [4].

Unlike applications described above, the ANOTIVE system has the following improvements:

- it focuses on any notifications that may appear on users' phones from different installed applications, messaging, or calls. Instead, in the applications described above, call filtering is more focused on different options, such as blocking calls from unknown numbers, private numbers, phonebook numbers, from the list of allowed or non-allowed numbers;
- in the messaging functionality group, we added the option to block a message that contains a specific word or wordlist;
- in addition, it also provides the functionality of answering to a notification, if possible;
- it does an adaptive filtering of calls and notifications, depending of the driving conditions.

There are also other apps for controlling notifications on the Android operating system [20], but we didn't find reports, in the literature, of apps that take over control over notifications received, taking into account driving contexts.

CONCLUSION AND FUTURE WORK

We introduced an in-vehicle software system that provides an adaptive filter of the calls and notifications launched by the driver's or co-driver's smartphone during driving.

The principle of "separation of concerns" used in the system development process allows other concerns to be dealt with in further development of the system in order to obtain future versions. For example, the system could automatically filter notifications into adaptive mode if the driver engages in overtaking another car.

ACKNOWLEDGMENTS

This research was funded by the Ministry of Research and Innovation of Romania, PN-III Program, Complex Project PN-III-P1-1.2-PCCDI-2017-0917, "Hybrid Light Visible Communication Platform and Augmented Reality for the Development of Intelligent Systems for Active Assistance and Safety of Vehicles", contract no. 21 PCCDI / 2018.

REFERENCES

1. Android Studio: https://developer.android.com/studio/
2. Astah IDE: http://astah.net/

3. Bazilinskyy, P. et al. Take-over requests in highly automated driving: A crowdsourcing survey on auditory, vibrotactile, and visual displays. *Transportation research part F: traffic psychology and behaviour* 56 (2018): 82-98.

4. “Blocare apeluri” application :

https://play.google.com/store/apps/details?id=com.vladlee.callblocker&hl=ro, access date: 3.05.2019

5. Bogdan, C., Serbanati, L. D. Toward a Concern-Oriented Analysis Method for Enterprise Information Systems. In *Proceedings of the IEEE International Multi-Conference on Computing in the Global Information Technology*, IEEE Computer Society (2006), 30-35.

6. Calls Blacklist - Blocați apeluri și SMS-uri: https://play.google.com/store/apps/details?id=com.vladlee.easyblacklist&hl=ro, access date: 2.05.2019

7. Console, L., et al. Personalized and adaptive services on board a car: an application for tourist information, *Journal of Intelligent Information System*s 21.3 (2003): 249-284.

8. Gamma, E., Helm, R., Johnson, R., Vlissides, J. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison Wesley Professional, 1994.

9. Larman, C. (2004). *Applying UML and Patterns. An Introduction to Object-Oriented Analysis and Design and the Unified Process*, Prentice Hall.

10. Maradona, R., et al. Adaptive behavior selection for autonomous vehicle through naturalistic speed planning. IEEE 20th *International Conference on Intelligent Transportation Systems* (ITSC). IEEE, 2017.

11. Marhan, A.M. *Evaluarea sistemelor interactive. Introducere în interacţiunea om-calculator* (coord. C. Pribeanu), Editura Matrix Rom, Bucureşti (2003): 179-200.

12. Moret, B. M. *The Theory of Computation*, Addison-Wesley, 1998

13. Notification Blocker & Cleaner & Heads-up Off application: https://apkpure.com/nl/notification-blocker-cleaner-heads-up-off/co.easy4u.ncleaner, access date: 2.05.2019

14. OMG: Unified Modelling Language Superstructure, version 2.0, ptc/03-0802, 2003.

15. Ossher, H., Tarr, P. Using multidimensional separation of concerns to (re)shape evolving software, *Communications of the ACM*, vol. 44, nr. 10, 2001

16. Pircalabu, V., 2018 (May). Mediafax.ro: https://www.mediafax.ro/externe/breaking-accident-grav-in-ungaria-in-care-a-fost-implicat-unmicrobuz-inmatriculat-in-romania-noua-persoane-au-murit-foto-17226243, access date 9.06.2019

17. Pressman, R.S. *Software Engineering: A Practitioner's Approach*, 8/e McGraw-Hill, 2014

18. Pribeanu, C. Un model detaliat al utilizabilitatii sistemelor interactive. *Revista Informatică Economică* 10 (1999): 31-36.

19. Robbins, N.B., Heiberger, R.M. (2011). Plotting Likert and Other Rating Scales" (PDF). JSM Proceedings, Section on Survey Research Methods. American Statistical Association. pp. 1058–1066, http://www.asasrms.org/Proceedings/y2011/Files/300784_64164.pdf

20. Sinha, R., 7 Best Smart Notification Apps for Android, https://beebom.com/smart-notification-apps-android/, access date: 3.05.2019

21. SQLite : https://www.sqlite.org/index.html, access date: 12.04.2019

22. Texting and Driving Accident Statistics, Edgar Snyder & Associates - A Personal Injury Law Firm Representing Injured People: https://www.edgarsnyder.com/car-accident/cause-of-accident/cell-phone/cell-phone-statistics.html, access date 9.06.2019

23. Wang, W., Xi, J., and Chen, H. Modeling and Recognizing Driver Behavior Based on Driving Data: A Survey, *Mathematical Problems in Engineering*, vol. 2014, Article ID 245641, 20 pages, 2014. https://doi.org/10.1155/2014/245641.

24. Wiese, E. E., and Lee, J.D. Auditory alerts for in-vehicle information systems: The effects of temporal conflict and sound parameters on driver attitudes and performance. *Ergonomics* 47.9 (2004): 965-986.