Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 3 (2015) 2658 – 2665

Procedia MANUFACTURING

6th International Conference on Applied Human Factors and Ergonomics (AHFE 2015) and the Affiliated Conferences, AHFE 2015

App analytics: predicting the distraction potential of in-vehicle device applications

M. Krause[a,]*, A.S. Conti[a], , M.Henning[b], C.Seubert[b], C. Heinrich[c], K. Bengler[a], C. Herrigel[d], D. Glaser[e]

[a]*Institute of Ergonomics, Technische Universität München, Boltzmannstr. 15 – Garching, Germany 85747*
[b]*Carmeq GmbH Berlin, Germany*
[c]*Daimler AG, Stuttgart, Germany*
[d]*Laudenklos Engineering Automotive, Germany*
[e]*GM Corporation, Detroit, USA*

Abstract

Three experiments were conducted to check the feasibility of predicting experimental outcomes of driver distraction studies. The predictions are based on subtasks analysis and synthesis. In the first experiment, data (e.g., Total Glance Time, Single Glance Durations and Total Shutter Open Times) are gathered when subjects interacted with touch screen applications. In a second experiment, additional data were gathered about rotary knob interactions. These data were used to synthesis and predict the outcomes of a third (evaluation) experiment, which involved rotary knob and touch screen tasks. The results are promising and can help to have a better understanding of problematic subtasks and reduce testing of clearly unsuitable applications. The transferof the procedure to other laboratories is challenging. The modeling and mapping process includes many subjective decisions.

Peer-review under responsibility of AHFE Conference

*Keywords:*Mobile devices;Subtask analysis; Synthesis; Driver distraction; Prediction; Estimate; Occlusion

* Corresponding author. Tel.: +49-89-28915404; fax: +49-89-28915389.
E-mail address: krause@tum.de

Peer-review under responsibility of AHFE Conference
doi:10.1016/j.promfg.2015.07.627

1. Introduction

As the development of applications (apps) for in-vehicle use becomes more prevalent, the investigation of these apps's driver distraction potential becomes increasingly important.This growing trend of using apps to substitute in-vehicle device functionality is not without cause—such applications have clear advantages: they are easily personalized, generally available at low cost, and functionality is familiar to the user. Disadvantages of developing apps for in-vehicle use is that, unlike OEM applications, 3rd-party apps typically have not been tested while driving to gauge if the app meets current industry driver distraction guidelines (e.g., AAM [1], ESoP [3], JAMA [5]).From the vantage point of the app developer, testing is logistically impractical. A developer would need to learn protocols for human-subject testing recruitment, purchase and house testing equipment, pay participants for their time, and spend many hours collecting and analyzing data (or pay a testing house to perform these services). Yet, understanding whether an app meets industry criteria remains critical. Since there are a vast number of apps on the market and even more to come that may be used while driving, a valid, efficient, and inexpensive method of evaluation has a clear need.

A practical starting point to develop the said method is to examine how common mini tasks (subtasks) within a larger task independently affect driver distraction metrics. Gaining this subtask knowledgewill allow a developer to more accurately predict whether a task, as a whole, will meet driver distraction criteria. The segmentation of a task into subtasks is called task analysis.In general, task analysis helps to explicitly answer the questions "*who* does *what* and *why*" ([4], p. 245). Systematic task analysis approaches such as GOMS [2] have already been developed and are well documented in empirical literature. GOMS is a predictive model approach to analyzing tasks and system interactions [2, 6]. There are many versions of GOMS with different foci; relevant to the current set of experiments is KLM-GOMS (Key-Level Model), which reduces tasks to a keystroke level [7]. Essentially, each subtask is assigned an operator, indicating the subtask type, and an execution time, which will be summed to acquire the total execution time needed to complete the entire task (see [7] for details regarding this procedure). This procedure has been documented for some mobile phone tasks [8] and has been adapted to some in-vehicle device tasks [9]—primarily navigation and route guidance tasks, while the vehicle is at a standstill.

In an effort to determine how subtasks contribute to industry driver distraction criteria, 3 experiments were conducted. The first two experiments were used to collect data on the unique contribution of various subtasks on driver distraction. As this project chose to adhere to the AAM Guidelines [1], the suggested driving performance, eye-tracking and occlusion measures were recorded.The third experimentwas conducted to establish whether the subtask data collected from experiments 1 and 2 were able to accurately predict glance related metrics of novel apps that share similar subtasks.

2. Methods

2.1. Design

This project was divided into three experiments:

- Experiment 1: Collect data (AAM driving task performance, eye tracking, occlusion) with **touch screen**
- Experiment 2: Collect data as in first experiment with **rotary knob**
    - Summarize results of first and second experiment and predict/estimate outcome of third experiment
- Experiment 3: Test apps with similar **touch screen**and**rotary knob**tasks
    - Comparepredicted outcome to the actual third experiment results

Table 1.Participants.

Experiment	N	male /female	Age min – max; M (SD)	Left handed	Eye ware needed	<10,000 km/year	>20,000 km/year
1	21	11/10	45 – 64; 56 (6)	1	18	3	7
2	21	11/10	46 – 68; 59 (6)	1	17	3	8
3	21	10/10	47 – 65; 55 (4)	2	17	5	7

2.2. Participants

All participants had a valid driver license.

Test persons reported on a five point Likert scale (never '1'-often '5') their usage of different devices (tablet, phone, satellite navigation systems [satnavs], pc, car). Most participants reported frequent use of phones and satnavs, however, 3subjects in the first experiment reported low usage of all devices. In experiment 2, 7 persons had no previous experience with a rotary knob. Subjects from the first two experiments were not allowed to participate in the third experiment. The usage experience of touchscreens in the third experiment was comparable to the group of experiment 1. Fifteen test persons had no previous experience with a rotary knob.

2.3. Apparatus and devices

All experiments were carried out in a static vehicle simulator with 55"-LCD for the driving scene. SILAB 4 (WIVW GmbH, Würzburg) was used to produce the driving simulation. The straight track resembled the AAM driving task specification and was adapted to Autobahn specifications for German drivers. The task was to follow a leading vehicle travelling at a constant speed of 80km/h at a safe distance of 50 meters. For the occlusion, PLATO spectacles (Translucent Technologies) were used with the system-paced protocol according to AAM (1500ms open; 1000ms closed). Eye-tracking was realized with a head-mounted Dikablis/D-Lab 2 system (Ergoneers GmbH, Manching).

Experiment 1 and 3 tested touch screen apps, which were displayed on anIntenso Tab 824 adjusted to an 800x480 display resolution (160ppi). Not used display areas were covered with a thick plastic shield. The tablet was mounted (swivelling)in front of a car radio (Sony CDX-GT570UI); see Figure 1a. The radio and tablet had a viewing angle of 30° below line of sight. Experiment 2 and 3 tested apps that required a rotary knob. For the rotary setup, a Daimler COMAND-system and coupling with Digital DriveStyle App on an iPhone 4 was used. The infotainment screen was mounted above the radio (see Figure 1b).

a b

Fig.1. (a) Touch screen setup; (b) rotary knob setup.

2.4. Tasks/subtasks

In the first experiment, 13 tasks were performedwith the touch screen. One task was later discarded due to hardware problems (poor touch recognition).Thus, 12 tasks (including 35 subtasks) qualified for the final analysis.In the second experiment, 10 tasks (with 33 subtasks) were performed with the rotary knob. In the third experiment,9 tasks were performed; 5 on the touch screen, 4 with the rotary knob. One touch screen task was later discarded due to hardware problems (poor touch recognition). The third experiment was not further split or analyzed on a subtask level.

In the third experiment, 1 task from the first experiment (touch screen) and 1 task from the second experiment (rotary knob) were replicated to test repeatability and variations. Additionally, all three experiments involved a radio tuning task on a hardware radio as a reference task, which is not included in the task count above or split into subtasks.

A single subtask can be involved in different tasks (e.g., “confirm a pop-up”). Typical examples for subtasks are: “enter a phone number”, “adjust a slider”, and “select a name from a list”.

2.5. Procedure

The subjects were acclimated to the simulation by driving at least three minutes on the straight track. Two minutes of baseline driving were then recorded. Each of the three experiments involved two measurement procedures: performing a task with 1) occlusion and 2) while driving with eye tracking. The order of the tasks (see section Tasks/Subtasks) within each experiment was randomized. Each task was explained and demonstrated separately to the subject by the experimenter. As soon as one task was demonstrated, the subject then performed iton their own, 1 time with the driving task and 1 time without. This phase was considered the task training phase. The testing phase for each task took place immediately after the task had been trained. Participants performed each task 4 times: twice with occlusion and twice with eye-tracking and driving, the order of which was alternated across experiments. The subjects were instructed to prioritize safe driving, when driving was part of their task set. The first experiment lasted approximately 3.5 hours, while the second and third experiment lasted around 2 hours each.

2.6. Metrics

The experimental data for the first and second experiment were split into subtasks by manual coding. The codes were set for the eye tracking data within the analysis program D-LAB 2.1 (Ergoneers GmbH, Manching) and for the occlusion with Interact 9 (Mangold International GmbH, Arnstorf) based on video recordings. The start of a subtask was directly at the end of the previous subtask. With frame-wise stepping, characteristic and perceivable display changes were manually searched for (e.g., the end of the subtask ‘*select app icon*’ is the appearance of the app’s startup screen and the start of the subtask ‘*startup delay*’). The task start/stop was visually encoded into the eye tracking and other video recordings by a button press executed by the experimenter, which lit anLED visible in the recordings.

After the first experiment, the data where split into subtasks and we tried to compose the experimental outcome of the whole tasks by their subtaskvaluesas a verification. We looked at three variables: Number of Glances (NoG), mean Single Glance Duration (mSGD) and mean Total Glance Time (mTGT). For NoG we found an average increment of 27% (SD 9%-point), for mSGD adecrement of -20% (SD 6% points) and for mTGT 0% deviation. This revealed a problem:the eye tracking analysis splits the gazes artificially. In Figure 2 this issue is depicted, where the hypothetical subject glances 3times, each for 1 second. The analysis divides these 3 glances into 4 glances and would calculate amSGD for Subtask1 of (1s+0.3s) / 2glances = 0.65s and for Subtask2 (0.7s+1s)/2glances = 0.85s. Therefore, we exported the data from the analysis software and coded our own work-around in Matlab.Thisis illustrated in Figure 2b: the number of glances for the subtask is calculated proportionally. Thus Subtask1 has 1.3 glances and Subtask2 1.7 glances. The mSGD is than calculated by the subject’s TGT divided by the subject’s NoG. In the example: for Subtask1 mSGD = TGT 1.3s / 1.3 glances = 1 second. These individual values from each subject were then averaged to calculate a mean value for each subtask. Based on thesevalues, a composition of the tasks from subtask for experiment 1 revealed an decrement of the composed NoG of 3% (SD 1%-point) and for the mSGD an increment of 11% (SD 2%-points). The task’s mSGD were not simply averaged from subtasks, else a weighted mean was calculated (for weighting the subtask NoG was used).For the occlusion the Total Shutter Open Times (TSOT) were derived by modulo calculations referenced to an initial shutter open event and with consideration of the subtask start/stop events. The shutter open/close eventswere video recorded via an LED in the camera's field of view.

Later we faced the challenge that for some metrics the 85th percentiles are of interest rather than mean values. We called our proposed solution to this problem the Berlin-Munich-Method: Instead of using a single averaged value to qualify each subtask (e.g., TGT for Subtask_X = Y seconds), we calculated an average value for each subject, thus, we have e.g., a data set TGT for Subtask_X = {Y_1 for subject1; Y_2 for subject2;…}. With this dataset, we ran a kind of 'virtual experiment'. In a selft-made excel tool, subtasks were able to be selected and were composed on a per subject basis to a complete subject task value. From these individual task values, mean and 85th percentiles were calculated. The currently composed values are TSOT, TGT, NoG which are summed up from subtask values and SGD, which is calculated by a weighted mean (weighting based on NoG). The estimations for experiment 3 were made with this tool and method.

3. Results and discussion

Due to a technical problem in the first experiment, the occlusion dataset was reduced from N=21 to N=14 subjects. The results of the third experiment are presented and compared to predictions based on the subtask data from experiment 1 and 2. Table 2 gives an overview of the tasks in the third experiment and how many subtasks were used to model them. The diagrams show the prediction of some metrics and the measured metric. The percent differences in the diagrams is related to the measured value. Thus, the difference between the estimated and actual measured values is divided by the measured value. For calculating the 85th percentiles, the interpolating Excel function was used.

a b

Fig. 2. (a) split glance problem; (b) proportional summing of number of glances.

Table 2. Experiment three task overview .All tasks included the starting of an application or entering of a submenu from a home screen.

Short Name	Device	Count of Subtasks	Short Description
T1	Touch	8	Enable a checkbox in a configuration submenu of an app and leave application
T2	Touch	2	Enter a calculation into calculator (about 10 input steps)
T3	Touch	5	Record a short voice message/note (one word)
T4	Touch	7	Search radio stream by text search (6 chars)
R1	Rotary	9	Share your location (Glympse) with someone from the contact list
R2	Rotary	10	Reply to a message with a predefined short text
R3	Rotary	3	Switch off the infotainment screen
R4	Rotary	5	Search a radio stream by text search (2 chars)

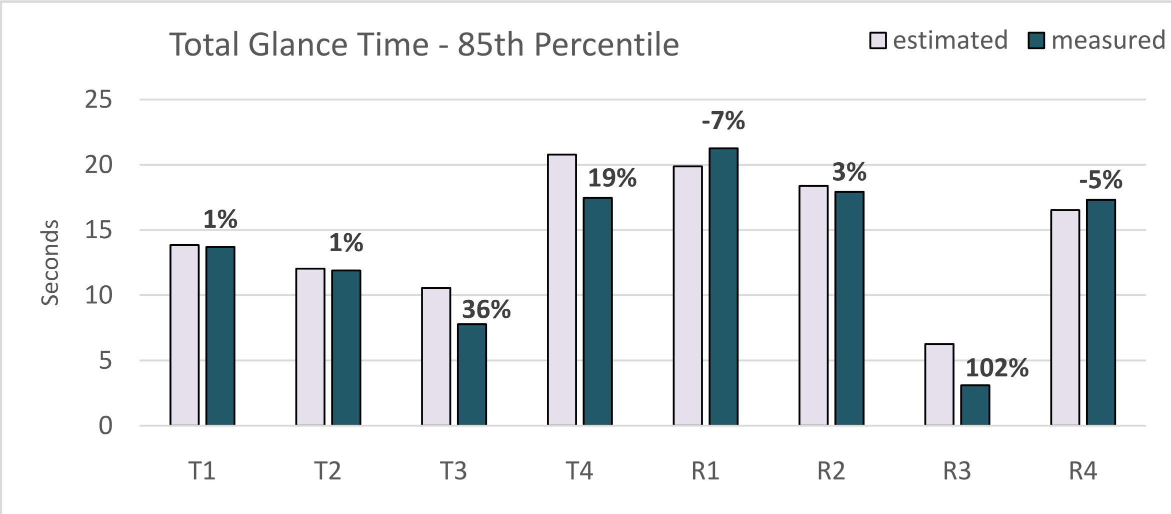

Fig. 3. Comparison of 85th percentile of Total Glance Time (TGT); estimated versus measured.

In Figure 3, the 85th percentiles of the estimated and measured Total Glance Times (TGT) are shown. Overall the estimations look satisfactory. Two deviations are obvious (T3, R3); these are the two shortest tasks. In task R3, the test subjects should switch off the infotainment screen, which required a short sequence of rotary knob interactions. Due to the short length and the haptic feedback of the rotary knob, it would have been possible to perform it blindly or with a few check glances. The task modeling uses subtasks that were extracted from longer, more complex interaction sequences. This may have contributed to the large overestimation.

Figure 4shows the Total Shutter Open Time(TSOT) from the occlusion method. Overall, similar results as in Figure 3 can be found. This is a good plausibility check, as the TSOT should reflect the TGT. Again R3's prediction has the highest overestimation.

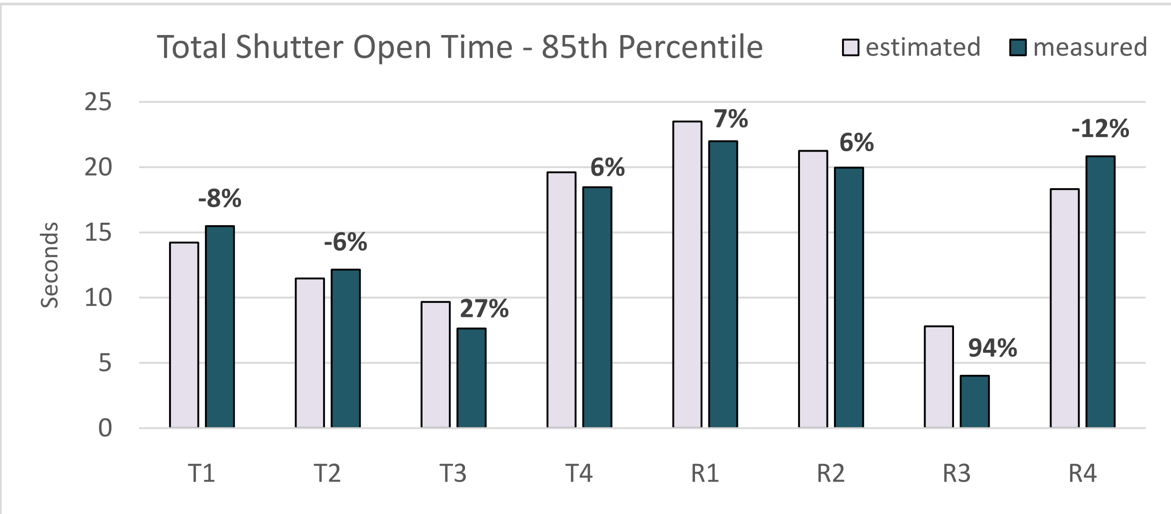

Fig. 4. Comparison of 85th percentile of Total Shutter Open Time (TSOT); estimated versus measured.

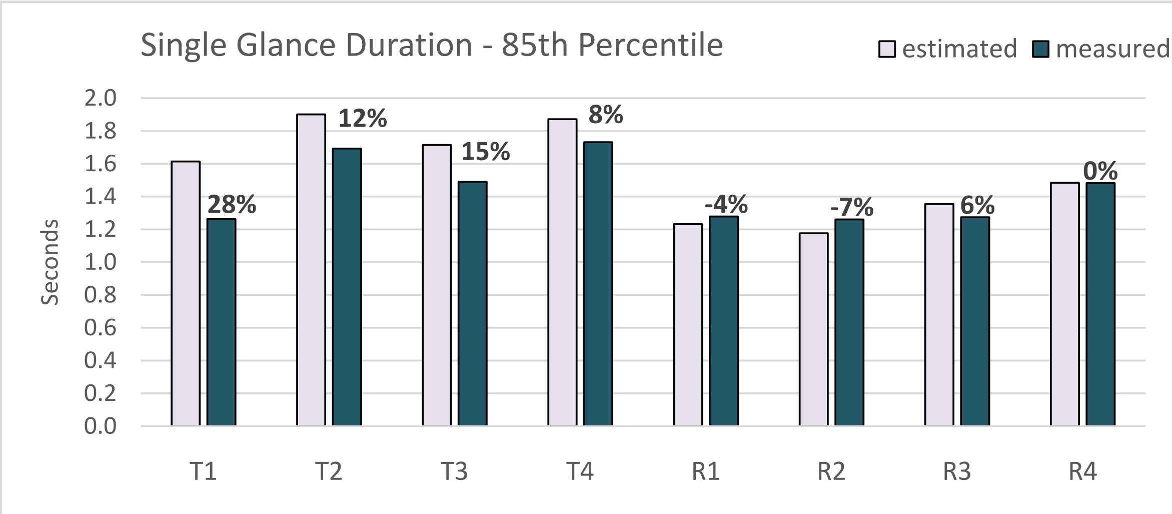

Fig. 5. Comparison of 85th percentile of Single Glance Duration (SGD); estimated versus measured.

The 85thpercentile of Single Glance Durations (SGD) are presented in Figure 5. The deviations are smaller than in the figures presented before. This is not surprising. If one assumes that a result for SGD will most likely between 1.2s and 2.0s, this reduced range leads automatically to smaller percent deviations. However, there would also be subtasks in the subtask list that could potentially raise the SGD estimation above the 2s criteria. As can be seen in Figure 5, there is an overestimation of all touch tasks. Nevertheless, for the rotary task group, the prediction would have been able to predict which task has the longest SGD (R4) and for the touch tasks the prediction would have right estimated the two tasks with the longest SGDs (T2, T4).

Experiment 3 included a replication of one task from experiment one (T1) and one from experiment 2 (R1). While the mean SGD for these show very similar results for the replication, the 85th percentile SGD for T1 is lowered by 0.13s compared to experiment 1 and R1 is increased by 0.18s compared to experiment 2. These can be indications that some subjects and data of experiment 3 might be affected by using rotary and touchscreen in one experiment (carry-over effect of glance strategies). Due to the fact that 85th percentilesare heavily influenced by a few datasets anyway, these also couldbe just artefacts.

T2 was modeled with just two subtasks (start the application and enter 10 inputs on a keyboard number pad). In this task, an equation with 10 entries is entered into a calculator interface. Here, the main subtask stems from a telephone task from experiment 1, in which phone numbers with 10 digits were entered. Because determining subcomponents of a task is somewhat subjective, it is often difficult to find an appropriate subtask (assuming it has already been tested). Among other things, these problems came up when we tried to transfer the method to further experimental results from other laboratories. So, the results presented here show a first step in this challenging process of synthesizing experimental results.

4. Conclusion

The results can be used in two ways. The list of subtasks with characteristic values (TGT, TSOT, SGD) could be a low-cost,valuable tool for developers looking to include different subtasks in their app and to estimate their distraction potential. It can be useful to make them more aware of driver distraction issues, thus, an educational aspect. Additionally, this could reduce the implementation of solutions that are unlikely to pass later driver distraction tests. Therefore, at very early phases of app development, more effortcan be directed to alternative solutions.

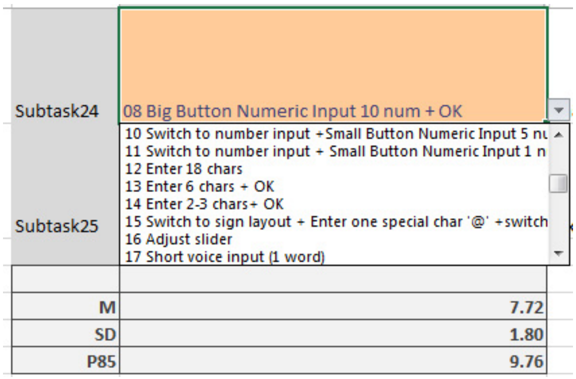

Fig.6. Excel tool to compose a task by subtasks and calculate the task metrics (in this case TGT).

A simple second excel sheet tool (see Figure 6) can be usedto compose an entiretask from subtasks via drop-down menus. This sheet then calculates and makes predictions based on the specified inputs (e.g. Berlin-Munich-Method, see section Metrics). At this point, these options are to be considered as useful prior to performing a driver distraction experiment with subjects in a laboratory rather than as a replacement for such. If apps become more popular within the vehicle, the laboratories equipped to test driver distraction would not be able to handle the vast amount of testing needed to keep pace with these implemented apps. An expert judgment about known principles (e.g., readability, user paced, no video, etc.), together with the excel sheet might be able to filter out clearly unreasonable and ill-suited apps that are not ready to even be tested with subjects for their driver distraction potential. Therefore, laboratory capacity can be reserved for more appropriate implementations to be tested.

Acknowledgements

The project was funded by the Car Connectivity Consortium (CCC). We would like to thank Nina Döllinger, Johannes Henel, Michael Hochwieser, Simone Kager, April Moeller, Thomas Moll, Lisa Scherer and Moritz Späth for supporting the experiments and analysis.

References

[1] AAM (2006) Alliance of Automobile Manufacturers. Statement of Principles, Criteria and Verification Procedures on Driver Interactions with Advanced In-Vehicle Information and Communication Systems. Driver Focus-Telematics Working Group
[2] Card, Moran, & Newell (1983). The Psychology of Human-Computer Interaction. Hillsdale, New Jersey: Lawrence Erlbaum Associates, Inc.
[3] ESoP (2006) European Statement of Principles. Commission Recommendation of 26 May 2008 on safe and efficient in-vehicle information and communication systems: update of the European Statement of Principles on human-machine interface (2008/653/EC)
[4] Hollnagel, E. (2012). Task Analysis: Why, What, and How. In G. Salvendy (Ed.), Handbook of Human Factors and Ergonomics (Fourth Edi., pp. 244–269). Hoboken, New Jersey: John Wiley & Sons, Inc.
[5] JAMA (2004) – Japan Automobile Manufacturers Association. Guideline for In-vehicle Display Systems — Version 3.0
[6] Olson, J. R., & Olson, G. (1990). The Growth of Cognitive Modeling in Human-Computer Interaction Since GOMS. Human-Computer Interaction, 5(2), 221–265. doi:10.1207/s15327051hci0502&3_4
[7] Kieras, D. (2001). Using the keystroke-level model to estimate execution times. *University of Michigan*. Last accessed: April 8th, 2015. Retrieved from http://www-personal.umich.edu/~itm/688/KierasKLMTutorial2001.pdf
[8]Schulz, T.W. (2008). *Evaluating Mobile Phones with the Keystroke-Level Model and other Desktop Methods* (Master's thesis). University of Islo, Department of Informatics.
[9] SAE; Society of Automotive Engineers (2002). Calculation of the Time to Complete In-Vehicle Navigation and Route Guidance Tasks (SAE Recommended Practice J2365, Warrendale, PA: Society of Automotive Engineers.