

*Proceedings of the
1st International Conference and Exhibition on Future RFID Technologies
Eszterhazy Karoly University of Applied Sciences and
Bay Zoltán Nonprofit Ltd. for Applied Research
Eger, Hungary, November 5–7, 2014. pp. 177–184*

doi: 10.17048/FutureRFID.1.2014.177

Defining metric to help environmental friendly use of mobile phones

David Zaborszky, Krisztina Erdelyi

John von Neumann Faculty of Information Technology, Óbuda University
david.zaborszky@gmail.com
erdelyi.krisztina@nik.uni-obuda.hu

Abstract

Nowadays the world is advancing toward such a direction, where the environmental impact is increasing, which can have long-term negative consequences for us and for the planet, when we can not solve this problem in a timely manner, sufficiently and efficient way. In order to reduce the pollution rate, we, IT equipment users can do many things to minimize the energy consumption. Suffice it to think of that, we should use our mobile devices more environmental friendly, which in this case means the purpose of achieving to consume less power.

Our goal is to develop a metric to calculate to what extent a mobile phone damages the environment regarding to the relevant user's habits. To achieve this, we examined the most popular mobile platforms so the metric is platform independent. A practical application based on the metric will be implemented for every popular mobile platform.

Keywords: environment, global, metric, mobile, IT, green, energy efficient

MSC: 68U35

1. The analysis of the platforms

In this chapter the main platforms will be exposed to multiple examinations to determine, what kind of data can they provide from themselves, and later these data can be used in the metric to perform the score calculations.

The platforms for the analysis will be the following:

- Android (API level 19)

- Windows Phone 8
- IOS 7
- Firefox OS (version 1.3.0)
- Sailfish OS (version 1.0.5.19)

1.1. The result of the platform analysis

The result of the platform analysis is the following [1][2][3][4][5][6][7][8][9][10][11]:

	Android	Windows Phone	IOS	Firefox OS	Sailfish OS
OS type of access	Open	Close	Close	Open	Open
Battery charge level	✓	✓	✓	✓	✓
Max capacity of the battery	✓				✓
Battery temperature	✓				
Current consumption					✓
WiFi network	✓	✓	✓	✓	✓
Mobile network	✓	✓	✓	✓	✓
Airplane mode	✓	✓	✓	✓	✓
Brightness	✓		✓	✓	✓
Automatic brightness	✓			✓	
Screen timeout	✓			✓	
Screen status	✓			✓	✓
Vibration	✓			✓	✓
Haptic feedback	✓			✓	✓
GPS module	✓	✓	✓	✓	✓
Volume	✓		✓	✓	✓
Bluetooth	✓	✓		✓	✓

1.2. Summary of the platform analysis result

The open source systems (like Android) provide more information about system modules, than the proprietary platforms (like Windows Phone or IOS). The following pieces of information can be accessed independently of the five analyzed platforms:

- The charge level of the battery

- The data connections: WiFi, mobile network
- The GPS

There can be defined an extended metric which will be platform independent within the open source systems. In this case the following pieces of information can be accessed for the metric:

- The charge level of the battery
- The data connections: WiFi, mobile network
- The GPS
- The Brightness
- The Vibration
- The Volume
- The Bluetooth

The determination of metrics will be discussed in the next section.

2. The determination of the metric

This chapter will focus on the definition of the metric.

2.1. Platform independent metric

The platform independent metric (indicated with P) will be calculated with the intersection of the information sources in the different examined systems.

2.1.1. Analysis for a given moment

The analysis for a given moment of the platform independent metric will be defined below. The WiFi adapter (indicated with P_W) is worth 50 point, when it is turned off, otherwise it will be 0 point. The mobile Internet (indicated with P_M) is worth 30 point, when it is turned off, otherwise it will be 0 point. The airplane mode (indicated with P_R) is worth 70 point, when it is turned on, otherwise it will be 0 point. The GPS module (indicated with P_G) is worth 50 point, when it is turned off, otherwise it will be 0 point. Thus during the analysis for a given moment methodology can be achieve minimum 0, and maximum 200 point.

The M^T is the theoretical case of the continuous metric score of the current platform independent analysis for a given moment, which can be calculated with the following (2.1) formula (n indicates the date of the end of the previous charge, and m indicates the date of the fully charge of the battery):

$$M^T = \int_n^m (P_W(t) + P_M(t) + P_R(t) + P_G(t)) dt \quad (2.1)$$

The M^P is the discrete metric score of the current platform independent analysis for a given moment (this discrete case can be used in the program), which can be calculated with the following (2.2) formula:

$$M^P = P_W + P_M + P_R + P_G \quad (2.2)$$

This method also provides a fast, but less accurate picture of the environmental pressure of the system.

2.1.2. Long-term analysis

The bases of the above discussed analysis for a given moment methodology will be used in the long-term analysis. The magnitude of the scoring are formed in the same way, but the allocation of the points are distributed proportionally depending on usage.

As time goes on, the points may increase, and the maximum score is not limited. So actually the most important factor will be also included in the calculation, which is the endurance of the battery in one charge. When the battery is charging, then the score has been reached its maximal value, and it can not increase further more, but it can still decrease, depending the overcharge of the battery. Therefore, when the overcharge is in progress, the system will decrease the score with a certain percentage (indicated with λ) in every hour. The value of the lambda is constant, and its definition is the following: $\lambda \in \mathbb{R}[0, 1[$

When the device is no longer in charging mode, then it reached its final score, which will be saved, and the counting begins again from 0.

The number of updates per hour (indicated with U_i) can be different. The i variable indicates the actual period. It can be easily calculated with the following method: Take the number of seconds in an hour (indicated with h , which is $h = 3600$), and divide it with the number of seconds since the last period. The T_i indicates the current time in seconds in the actual period, and T_{i-1} is the actual time in seconds in the previous period. The U_i can be calculated with the following (2.3) formula:

$$U_i = \frac{h}{T_i - T_{i-1}} \quad (2.3)$$

The i indicates the periods ($i \in \mathbb{N}, i = 1, \dots, n$). If it is ensured that the different periods are activate in the same intervals, then the U_i can be a constant.

The overcharge multiplier in continuous case (indicated with O^T) can be calculated in the following way (2.4) (c indicates the overcharging time in hours $c \in \mathbb{R}_0^+$):

$$O^T = (1 - \lambda)^c \quad (2.4)$$

The M^T is the theoretical case of the continuous metric score of the current platform independent long-term analysis, which can be calculated with the following (2.5) formula (n indicates the date of the end of the previous charge, and m indicates the date of the fully charge of the battery):

$$M^T = \left(\int_n^m (P_W(t) + P_M(t) + P_R(t) + P_G(t)) dt \right) * (1 - \lambda)^c \quad (2.5)$$

The overcharge multiplier in discrete case (indicated with O^D) can be calculated in the following way (2.6) (c_{sec} indicates the elapsed time in seconds $c_{sec} \in \mathbb{N}$):

$$O^D = (1 - \lambda)^c = (1 - \lambda)^{\left(\frac{c_{sec}}{h}\right)} \quad (2.6)$$

The c can be calculated until j period in discrete case with the following formula (2.7) ($j \in \mathbb{N}$ and T_j indicates the date of the last overcharge period):

$$c(j) = T_j - T_0 = c(j - 1) + (T_j - T_{j-1}) \quad (2.7)$$

In discrete case, p is the time of the one hour after the first overcharge period, which can be calculated with the following (2.8) formula:

$$p = T_0 + h \quad (2.8)$$

The M^P is the current platform independent long-term analysis discrete metric score, which can be calculated with the following (2.9) formula:

$$M^P = \left(\sum_{i=0}^{\infty} \left(\frac{P_{W_i} + P_{M_i} + P_{R_i} + P_{G_i}}{U_i} \right) \right) * \left((1 - \lambda)^{\left(\frac{T_j - p}{h}\right)} \right) \quad (2.9)$$

The i indicates the periods without overcharge ($i \in \mathbb{N}, i = 1, \dots, n$), and the j indicates the periods with overcharge ($j \in \mathbb{N}, j = 1, \dots, n$).

This method also provides a slower, but a more accurate picture of the environmental pressure of the system.

2.2. Extended metric

The extended metric (indicated with N) is now only supports the open source operating systems, so in this case, it is platform independent between the open source platforms. It will be calculated with the intersection of the information sources in these kinds of examined systems. This is actually the expansion of the previously discussed platform independent metric.

2.2.1. Analysis for a given moment

The above described platform independent factors (P_W, P_M, P_R, P_G) will be the same, but there will be additional factors in this metric. The Bluetooth adapter (indicated with N_B) is worth 50 point, when it is turned off, otherwise it will be 0 point. The vibration (indicated with N_V) is worth 5 point, when it is turned off, otherwise it will be 0 point. The volume size is inversely proportional to the maximum of 5 points (indicated with N_H). For example, if the volume is turned down completely, then it will be 5 points. If it is 50% then it will be 2.5 point. If it is maximum, then it will be 0 point. The brightness (indicated with N_F) is worth 40 point, when it is in automatic mode, otherwise it is inversely proportional to the maximum 40 point. For example, if the brightness is not in automatic mode and its value is at maximum, then the point will be 0. If it is in 50%, then it will be 20 point.

The M^T is the theoretical case of the continuous metric score of the current platform independent analysis for a given moment, which can be calculated with the following (2.10) formula (n indicates the date of the end of the previous charge, and m indicates the date of the fully charge of the battery):

$$M^T = \int_n^m (P_W(t) + P_M(t) + P_R(t) + P_G(t) + N_H(t) + N_B(t) + N_V(t) + N_F(t)) dt \quad (2.10)$$

The M^P is the discrete metric score of the current extended analysis for a given moment, which can be calculated with the following (2.11) formula:

$$M^P = P_W + P_M + P_R + P_G + N_H + N_B + N_V + N_F \quad (2.11)$$

This method also provides a fast, but less accurate picture of the environmental pressure of the system.

2.2.2. Long-term analysis

The bases of the above discussed analysis for a given moment methodology will be used in the long-term analysis. In this method, the determination of the score will be the same way as in case of the platform independent metric, but it will be also extended. It will take more possibilities into consideration.

The points will be awarded proportionally depending on the usage, like in case of platform independent metric. The screen (indicated with N_F) is worth 50 point, when it is turned off in the actual period, otherwise it will be 0 point.

The M^T is the theoretical case of the continuous extended metric score of the current long-term analysis, which can be calculated with the following (2.12) formula (n indicates the date of the end of the previous charge, and m indicates the date of the fully charge of the battery):

$$M^T = \left(\int_n^m (P_W(t) + P_M(t) + P_R(t) + P_G(t) + N_H(t) + N_B(t) + N_V(t) + N_F(t) + N_K(t)) dt \right) * (1 - \lambda)^c \quad (2.12)$$

The M^P is the current extended long-term analysis discrete metric score, which can be calculated with the following (2.13) formula:

$$M^P = \left(\sum_{i=0}^{\infty} \left(\frac{P_{W_i} + P_{M_i} + P_{R_i} + P_{G_i} + N_{H_i} + N_{B_i} + N_{V_i} + N_{F_i} + N_{K_i}}{U_i} \right) \right) * \left((1 - \lambda) \left(\frac{T_j - p}{h} \right) \right) \quad (2.13)$$

The i indicates the periods without overcharge ($i \in \mathbb{N}, i = 1, \dots, n$), and the j indicates the periods with overcharge ($j \in \mathbb{N}, j = 1, \dots, n$).

This method also provides a slower, but a more accurate picture of the environmental pressure of the system.

Conclusion

We have defined two metrics in the above. The first one is a fully platform independent metric, which provides a consistent, comparable result for the measurement of the environmental load of the mobile device. The second one is the extended metric, which provides a uniform, comparable and extended result in the open source platforms, for the same reason as the other metric. So actually the extended metric provides a platform independent result in case of the open source systems.

There is an application in the most common platforms (Android, IOS, Windows Phone), which uses these metrics, and a web application where the users can compare their points to the others.

References

- [1] EKLER PÉTER, FEHÉR MARCELL, FORSTNER BERTALAN, KELÉNYI IMRE, Android-alapú szoftverfejlesztés, *SZAK Kiadó Kft* (2012).
- [2] WALLACE JACKSON, Learn Android App Development, *Apress* (2013).
- [3] ÁRVAI ZOLTÁN, FÁR ATTILA GERGŐ, FARKAS BÁLINT, FÜLÖP DÁVID, KOMJÁTHY SZABOLCS, TURÓCZY ATTILA, VELVÁRT ANDRÁS, Windows Phone fejlesztés lépésről lépésre, *Jedlik Oktatási Stúdió Kft* (2012).

-
- [4] TOMASZ SZOSTAK, Windows Phone 8 Application Development Essentials, *Packt Publishing* (2013).
 - [5] JACK NUTTING, FREDRIK OLSSON, DAVID MARK, JEFF LAMARCHE, Beginning iOS 7 Development: Exploring the iOS SDK, *Apress* (2014).
 - [6] ROB NAPIER, MUGUNTH KUMAR, iOS 7 Programming Pushing the Limits, *Wiley* (2014).
 - [7] MICROSOFT CORPORATION, Develop universal Windows apps, <http://dev.windows.com/en-us/develop>, (Accessed: 14 April 2014).
 - [8] GOOGLE INC., Android develop, <http://developer.android.com/reference/packages.html>, (Accessed: 22 April 2014).
 - [9] APPLE INC., iOS Developer library, <https://developer.apple.com/library/ios/navigation/>, (Accessed: 26 April 2014).
 - [10] MOZILLA CORPORATION, Firefox OS, https://developer.mozilla.org/en-US/Firefox_OS/, (Accessed: 1 May 2014).
 - [11] JOLLA LTD., Sailfish OS Develop overview, <https://sailfishos.org/develop-overview-article.html>, (Accessed: 1 May 2014).