Comparing Energy Profilers for Android

Alexander Bakker
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
a.bakker-5@student.utwente.nl

ABSTRACT

Energy profiling is the analysis of the energy usage of applications. This can be useful for developers aiming to make their applications more energy efficient. An energy profiler is an application that can aid in this analysis. There are multiple energy profilers available. There is however, to our knowledge, no literature comparing Android energy profilers, and no survey of papers on this subject.
In this paper, an analysis is given of the functionalities that a set of energy profilers offers, and a comparison is made between these profilers on several areas.

Keywords

Android, Energy Profiling, Energy Efficiency, Measurement, Power Monitoring, Battery Life

1. INTRODUCTION

Android smartphones are getting increasingly more powerful, and with that, the energy consumption of smartphones is steadily going up. There has, however, not been a major breakthrough in battery capacity for years, and such breakthroughs are not expected for years to come [5]. Unacceptable battery use can be a reason for users to uninstall applications, or give negative feedback on them. There are several components in smartphones that consume energy, such as the CPU and the Wi-Fi adapter. Developers will want to know how much their application uses these components.

In order to provide a positive user experience and develop energy-efficient applications, developers need to monitor the energy consumption of their applications, in order to further optimise them.

If a developer wants to know how much energy their application uses, one way of finding out would be to outfit a device with current and voltage meters and physically measure the energy that it uses. There are several drawbacks to this method, the main drawbacks being that it is far from trivial, and that the resulting numbers do not reflect the usage of the one application that the developer wants to profile, but rather the usage of the device as a whole.

21st Twente Student Conference on IT June 23rd, 2014, Enschede, The Netherlands.

Another approach to profile applications is to use so-called energy profilers, which are applications that use the battery level sensor (an internal voltmeter) to calculate the energy that an application uses. The main advantage of this approach is that it requires far less effort from the developer's side. Höpfner [3] discusses in greater detail the reasons for further exploring energy profiling.

This paper compares Android energy profilers and the functionalities they offer. It provides an overview of the main energy profilers which are publicly available, as well as energy profilers described in the literature. In addition, this paper will discuss which profilers are most useful for profiling Android applications.

Throughout this paper we will refer to applications that generate energy profiles as *energy profilers* or simply *profilers*, and to other applications as *applications*.

2. RELATED WORK

An attempt was made to find related work. Unfortunately, no such work was found. Work related to the subject of this paper would compare mobile applications, and could have been of interest in determining what criteria to compare profilers on. Some studies have been found on comparing the usability of mobile applications, but as usability is not the main focal point of this paper, these studies have not been referenced.

3. SEARCH METHOD

In this section we will outline what steps have been taken to search for profilers, and which profilers have been found. The first step was to conduct a literature review to establish what work has already been done in this field. Specifically we have searched Google Scholar and Scopus, and compiled a list of all papers that related to Energy Profilers. The search terms that were used are `Android AND ("power usage" OR "battery usage" OR "system") AND monitor`. The scope of this literature study was limited to energy profilers for the Android platform, since this is also the scope of this paper. During this literature search, seven profilers were found:

- PowerTutor [17]
- SEMO [1]
- BatTrace [9]
- eDoctor [8]
- AppScope [16]
- DevScope [6]
- Sesame [2]

Since profilers are themselves also applications, the second step was to search the Google Play store for profilers.

While searching the Google Play store, it became apparent that a lot of applications exist that are so-called *battery monitors*, which merely mirror the functionality of Android's internal battery monitor. Since these merely add a different graphical look and feel to functionality already present in the Android platform, all but one of these were excluded from the list. After this second study, we had a list of 13 profilers:

- PowerTutor [17]
- SEMO [1]
- BatTrace [9]
- eDoctor [8]
- AppScope [16]
- DevScope [6]
- Sesame [2]
- Intel Performance Viewer [15]
- Trepn [4]
- GSam Battery Monitor [14]
- Battery Log [11]
- BPU Monitor [12]
- Dr Power [13]

Additionally, an attempt has been made to search for energy profilers using Google, but this did not result in any additional profilers.

The third step was to filter this list. The profilers found in our literature study were often prototypes, which were either not downloadable or not ready for use. Since the purpose of this paper is to provide a comparison of usable applications, a lot of the applications found in the literature study were removed from the list.

These steps resulted in the following list of energy profilers for further study:

1. Intel Performance Viewer
2. PowerTutor
3. eDoctor
4. Trepn
5. GSam Battery Monitor
6. CPU Monitor

4. PROFILER DESCRIPTION

In this section we will give a short description of all profilers, outlining the key features and functionalities they offer. A more in-depth analysis of functionalities that the profilers offer can be found in the next section.

4.1 Intel Performance Viewer

Intel Performance Viewer [15] is a performance monitoring tool that collects information about the foreground application and displays it in a real-time graph. The graph is displayed as a semitransparent overlay graph. The profiler can be used for quick debugging, profiling, or for identifying resources an application is using. On rooted devices, the profiler can also gather low-level hardware performance data. Rooting is the acquiring of administrator privileges on a device, and may void the warranty of the device.

Intel Performance Viewer is unable to log the data it collects.

4.2 PowerTutor

PowerTutor [10] is an energy profiler for Android that was developed at the University of Michigan, with the help of Google. The key feature of PowerTutor is that it can measure the actual energy that applications consume through hardware, where most profilers are only able to give information about the usage of hardware components per application. PowerTutor is also able to list applications according to the amount of energy they use on different components (3G, Wi-Fi, CPU or screen) and can generate log files. It has several ways of visualizing the data, such as a graph that plots the amount of energy a component uses per time-unit, or a pie chart comparing the relative energy used by different components. PowerTutor can generate a list of all applications running on the phone, sorted on the relative amount of energy PowerTutor estimates these applications use. It can also visualise how much an application uses hardware components.

4.3 eDoctor

eDoctor is the only profiler in this paper that we were unable to download and test. The reason for including it in this study is because of the unique approach it takes to profiling applications. The information about eDoctor is extracted from *Automatically Diagnosing Abnormal Battery Drain Issues on Smartphones* [8].

eDoctor distinguishes itself from other profilers in that it specifically looks for *Abnormal Battery Drain* (ABD), which is battery drain that is not caused by normal resource usage. The reason for looking for ABD, rather than just high battery usage, is that high battery usage might be inherent in the application: one would expect a graphically intensive application such as a game to use a lot of energy, simply because it is a very heavy application.

eDoctor records resource usage and relevant events, such as version or configuration changes, and then uses this information to diagnose ABD. Additionally, eDoctor can suggest resolutions for ABD issues, such as reverting to a previous version or terminating the application after use.

4.4 Trepn Profiler

Trepn Profiler (or Trepn for short) [4] is an energy profiler developed by Qualcomm, for Android devices using Qualcomm Snapdragon processors. Trepn can help developers optimise code for CPU usage and frequency, memory statistics, and network usage. It can display data in real-time or store it in a log file for offline analysis.

Trepn offers a very customizable set of overlays, which can help with the real-time profiling of applications based on CPU, memory or network usage. It also offers detailed graphs that can help in analysing how applications use energy. Trepn can analyse one particular application, or the device as a whole.

Qualcomm also offers an Eclipse plug-in for Trepn, that developers can use to correlate battery spikes with system events. Using this plug-in, developers can import the data Trepn generates directly into Eclipse, which can make development more efficient.

4.5 GSam Battery Monitor

GSam Battery Monitor [14] is a battery monitoring application mainly designed for consumers, that also has energy profiling functionality. GSam offers several functionalities that consumers can use to help manage the energy usage of their device. It lists applications sorted by how much energy they use, or how much they use CPU, network or GPS. It does offer a graph, but this contains the same information as the graph that Android's battery monitor uses. It also offers a reskinned battery indicator, which displays the percentage of battery remaining in the notification bar, a useful feature for consumers currently not present in vanilla Android.

4.6 CPU Monitor

CPU Monitor [12] is a profiler specialised in analysing the CPU usage of applications. It can also analyse memory usage and network activity of the system as a whole and for individual applications, and plot these in a graph. An additional feature of CPU Monitor is that it can give an estimate of what applications slow down a device most, based on CPU and memory usage. It has a scheduling function that can start profiling on set moments, or when CPU usage exceeds a certain threshold for at least a certain amount of time.

5. FUNCTIONALITIES

In this section we will discuss what functionalities profilers offer. On a very low level, functionalities are almost exclusively binary: for instance, either a profiler is able to track CPU usage per application, or it is not. Since a table listing all functionalities for all profilers would become unnecessarily large and unreadable, we have decided to group functionalities in meaningful groups.

The most important group of functionalities is a list of what different profilers can measure from an application. First, we will discuss features from a *high-level* perspective. This encompasses general functionalities with respect to profiling. Second, since profiling of applications relies for a large part on being able to measure in what ways applications use hardware components, we will discuss which *hardware components* the selected profilers are able to profile.

Since the goal of profilers is to give their users insight in the data they collect, another important group of features is *data visualisation*.

Finally, some *general information* is given about the profilers in terms of their availability and other statistics. These are not so much functionalities of the profiler, but can nevertheless be important for users.

5.1 High-level functionality

The high-level functionalities of profilers are broadly defined as "what is the scope of things it can profile". An overview of high-level functionalities is given in Table 1. This section discusses these in more detail.

5.1.1 Systemwide

The function to profile hardware components system-wide means that a profiler can analyse the usage of hardware components irrespective of what application is using them. That is, the profiler can measure how much components are used in total. There are 4 profilers that offer functionality in this area. Intel Performance Viewer does offer functionality in this area, but the device needs to be rooted to be able to use it. PowerTutor's unique feature in this area is that it measures and plots the estimated amount of energy that is used on components system-wide, rather than how much components are actually used. Trepn and CPU Monitor are also able to measure how much components are used, but they do not estimate the amount of energy that is used.

5.1.2 Per application

The ability to profile individual applications is a very important feature for developers, as this can help them write applications that utilise resources more efficiently. All of the inspected profilers offer some functionality in this area.

Intel Performance Viewer, Trepn and CPU Monitor offer a comparable set of functionality in this area. All are able to track roughly the same set of data points. A developer can get a general idea of how much an application uses most of the hardware components in a device using any of these profilers.

eDoctor is also able to profile applications, but only to the extent that it will measure ABD. For this purpose, eDoctor uses maintenance, version and configuration information gathered over longer periods of time. If eDoctor notices a large increase in the amount of energy that an application uses between two versions, it concludes that between these versions something must have happened that caused the application to use more energy, and will advise the user on how to fix this issue. It is therefore not so much useful for developers (since it will only tell whether an application started using more energy), but it is interesting for consumers in this respect to keep their energy usage in check.

PowerTutor is a good profiler for determining the amount of energy that an application uses. It calculates an estimate of how much energy each application uses, and is able to break this down per component. It collects information about the energy usage of components, and builds a model with this information that it uses to calculate the energy usage per application.

GSam Battery Monitor can generate a list of all applications running on the phone, sorted according to the relative amount of energy they use. It can also calculate some data points about how an application uses components. However, it does not give a lot of insight in how it calculates energy usage, and is unlikely to lead to useful insights for developers. All other profilers that were investigated are better at generating useful data.

5.1.3 Online profiling

4 out of 6 profilers that were investigated are able to measure performance in real-time, and thus give insight in how a phone uses resources as a developer is testing it.

Intel Performance Viewer and Trepn are able to show overlay graphs displaying information on the usage of hardware components. Overlay graphs are graphs that can be displayed on top of other applications. This is useful, as it allows developers to look at the usage of certain hardware components as the application is being used.

Intel Performance Viewer is unique in the respect that it is able to show information about just the application that is running in the foreground. This makes it a good tool for detecting spikes in CPU or 3G usage while using the application. However, Intel Performance Viewer requires root access to profile the entire system. This does mean that it is less usable for systemwide profiling.

PowerTutor and CPU Monitor are also able to show real-time information about the phone, but do not have an overlay function. This means that one can only look at real-time information about the device from within these profilers.

5.1.4 Detect ABD

Abnormal Battery Drain, or ABD for short, is a concept that is introduced by Xiao et al. in their paper outlining eDoctor [8]. They define ABD as follows:

> ABD refers to abnormally fast draining of a smartphone's battery that is not caused by normal resource usage. From a user's point of view, the device previously had reasonable battery life under typical usage, but at some point the battery unexpectedly started to drain faster

Table 1. High-level functionality per profiler

	Per component	Per app	Online profiling	Detect ABD	Average energy use/app	Logging
Intel Performance Viewer	Yes	Yes	Yes	No	No	No
PowerTutor	Yes	Yes	Yes	No	Yes	Yes
eDoctor	No	Yes	No	Yes	No	Yes
Trepn	Yes	Yes	Yes	No	No	Yes
GSam Battery Monitor	No	Yes	No	No	No	Yes
CPU Monitor	Yes	Yes	Yes	No	No	Yes

> than usual. As a result, whereas users might comfortably and reliably use their phones for an entire day, with an ABD problem their batteries might unexpectedly exhaust within hours.

From all profilers, only eDoctor is able to analyse Abnormal Battery Drain. Other profilers offer no means of analysing ABD.

Since eDoctor has only been described in literature, but was unavailable for download, no further details about its functionality have been found. Furthermore, it is unknown what devices or versions of Android this profiler will run on.

5.1.5 Average energy use per application

Measuring the amount of energy an application uses is an important part in profiling. Most profilers offer an indirect way of inferring the amount of energy that applications use, or measure the total amount of energy used by an application. PowerTutor is the only profiler under consideration that is able to directly calculate the average amount of energy used by applications; that is, the total amount of energy used divided by the time the application has been running.

5.1.6 Logging

Some profilers are able to save their results in some way, for example in a log file or as a snapshot, for later analysis. Logging data can be useful for developers in many cases, as it enables them to look at the stored data in more detail, or at a later time.

PowerTutor is able to store a detailed power trace of all applications running on the device. It stores the energy consumption of each hardware component every one second, both in total and decomposed per application.

Even though PowerTutor only stores information related to energy consumption (rather than, for instance, CPU load or frequency), developers can use this information to gain valuable insight in the way their application consumes energy.

eDoctor was designed to run for longer periods of time, as it can only detect abnormal battery drain if it can compare an application's energy usage with that of a previous version. eDoctor stores this information in its app-specific storage, so it is not directly accessible for users. The developers of eDoctor state that all analysis of the data is done locally in eDoctor, but they do not explain in what way users can access this information [8].

When Trepn runs, it is able to store all information related to this run as either a `.csv` or a `.db` file. When storing a run in a `.csv` file, it is stored in a way that can easily be read by spreadsheet programs. Trepn is unable to read `.csv` files itself, so in order to be able to analyse a run within Trepn later on, the run has to be stored in the `.db` format.

GSam Battery Monitor will store and display a list of all applications and their estimated energy usage since the last time the phone was disconnected from the charger. Alternatively, users can choose to have GSam show the energy usage since the last time the phone was fully charged. There is no mention of how GSam stores the data it collects.

CPU Monitor can store all the information it collects in `.txt` files.

Intel Performance Viewer does not provide any means of storing results.

5.2 Component-level

In this section, we will discuss in more detail what hardware components each of the profilers are able to analyse. An overview of what components each profiler can analyse is provided in Table 2.

5.2.1 Intel Performance Viewer

Intel Performance Viewer is able to analyse CPU, memory, Wi-Fi/3G, and storage data. In terms of Wi-Fi/3G, it is able to measure how much kB's are being sent and received per application. It does not differentiate between Wi-Fi and 3G.

Its main focus appears to be storage-related analysis, as it is able to analyse Read Operations, Data Sectors Read, Read Time, Write Operations, Data Sectors Write, Write Time, and I/O time for both internal storage and external (microSD) storage. It can analyse these for the application running in the foreground. It can also analyse these data points systemwide, but in order to do this it requires root access.

5.2.2 PowerTutor

PowerTutor is able to analyse the energy used by applications on several components, and on components as a whole. For applications, it can analyse the energy usage in the CPU, display, Wi-Fi and 3G. It can also analyse the energy usage on audio and GPS, but can not trace these to individual applications.

5.2.3 eDoctor

eDoctor is able to track Abnormal Battery Drain from applications based on their CPU, 3G/Wi-Fi and GPS usage. Additionally, eDoctor can also analyse the amount of energy used by applications on audio, wakelocks (which is a lock implemented in Android that applications can use to prevent the screen from going to sleep) and various sensors.

GPS uses energy regardless of how many applications are using GPS data. When multiple applications are using GPS data at the same time, eDoctor divides the energy usage on GPS over these applications.

eDoctor is unable to perform systemwide profiling.

5.2.4 Trepn

Trepn is able to analyse the usage of CPU, 3G, Wi-Fi and GPS. Additionally, and unlike the other profilers under investigation, Trepn is able to analyse GPU (graphics

Table 2. Component-level functionality per profiler

	CPU	3G	Wi-Fi	GPS	Battery status	GPU
Intel Performance Viewer	Yes	Yes	Yes	No	No	No
PowerTutor	Yes	Yes	Yes	Yes	Yes	No
eDoctor	Yes	Yes	Yes	Yes	No	No
Trepn	Yes	Yes	Yes	Yes	Yes	Yes
GSam Battery Monitor	No	Yes	Yes	No	Yes	No
CPU Monitor	Yes	Yes	Yes	No	Yes	No

processing unit) usage, should a device have one. In terms of Wi-Fi and 3G Trepn can analyse the amount of data that is sent and received.

It is able to profile individual applications, or the system as a whole.

5.2.5 GSam Battery Monitor

GSam is able to measure the amount of data sent and received over 3G and Wi-Fi, but does not differentiate between these. It can also trace the amount of seconds that an application uses the CPU, but does not collect further CPU usage data. Additionally, GSam is able to track the number of wakelocks an application uses, and the usage of various sensors.

5.2.6 CPU Monitor

CPU Monitor is able to gather CPU, 3G and Wi-Fi information. It can also trace these to specific applications. However, it does not differentiate between data sent and received over Wi-Fi and over 3G.

CPU Monitor is able to also calculate the average usage of the CPU by applications, and to give a list of all processes sorted by their impact on performance.

CPU Monitor also has a setting that will enable it to start recording automatically when CPU usage is at or above a certain threshold for a certain period of time, or start profiling at specific moments of the day.

5.3 Data visualisation

An important aspect of these applications is the ways in which they can visualise the data they collect. Common visualisation methods are graphs, bar charts and pie charts. An overview of the visualisation methods can be found in table 3. In this section, we will discuss these in more detail.

5.3.1 Intel Performance Viewer

Intel Performance Viewer's only method of visualising data is by using a real-time transparent overlay with line graphs for each of the data points being traced. The graph can be resized, and the granularity of the graph can be set to any value between 100ms and 2000 ms. Using this graph, developers can inspect the usage of components by an application while it is being used.

Other than this, Intel Performance Viewer does not offer any visualisation.

5.3.2 PowerTutor

PowerTutor is able to visualise systemwide and application-specific information in three ways:

- Pie charts: both systemwide and application-specific information can be visualised in a pie chart.
- Line graphs: systemwide information can be visualised with line graphs, which show how much energy different components have used for the last minute.
- Ordered list: PowerTutor can generate a list of all applications running on the phone. At the top of the list are tabs labelled LDC, CPU, Wi-Fi and 3G; by toggling these on or off users can order applications based on the sum of all of these parts.

The visualisations PowerTutor offers give a functional, easy to use insight in how much each application uses on each hardware component. If developers need a more detailed analysis, it is also possible to generate a log file.

5.3.3 eDoctor

Since eDoctor is unavailable for download, we were unable to check what visualisations it offers.

5.3.4 Trepn

Trepn is able to generate graphs displaying information about CPU usage and frequency, memory usage and the amount of wakelocks. These graphs show systemwide information only.

Trepn can also display this information through customisable overlay graphs, so developers can get insight in the usage of resources as they're testing their application. For these overlays, Trepn offers three graph types: 'graph', 'volt meter' and 'bar chart'. These are three ways to visualise the same data. Of these, graph is the most useful, since it allows users to trace a data point over some time. Bar chart and volt meter are essentially one-dimensional ways of visualising a data point.

5.3.5 GSam Battery Monitor

GSam Battery Monitor is able to generate a line graph displaying the battery percentage over time. Also displayed in this line graph is GSam's estimate of how long the battery will hold out under current use. GSam can overlay this with 1) the battery temperature, 2) level of GSM signal, or 3) a view containing the activity of GPS, Wi-Fi, display activity, and duration of phone calls. These views all display systemwide information, and are therefore not useful for profiling applications. They can be useful for determining what components cause battery drain in the device. However, PowerTutor does a better job at this by actually measuring the amount of energy that different components use.

Apart from this, GSam is able to generate a list of applications according to the relative amount of energy that they consumed since the last time the device was charged. This is sufficient for comparing the energy use of different applications.

It is also able to display some component usage statistics per application. The usage statistics shown are not very useful for profiling applications, though.

5.3.6 CPU Monitor

CPU Monitor is able to generate line graphs showing CPU usage and frequency and memory and network usage. It can do this both systemwide, and for specific applications. For systemwide processes, it is also possible to see this

Table 3. Data visualisation functionalities per profiler

	Graphs per component	Graphs per application	Textual ('form')	Overlay graphs
Intel Performance Viewer	Yes	Yes	No	Yes
PowerTutor	Yes	Yes	Yes	No
eDoctor	No	No	No	No
Trepn	Yes	Yes	Yes	Yes
GSam Battery Monitor	No	No	Yes	No
CPU Monitor	Yes	Yes	Yes	No

information in real-time. Two points of criticism here are that the in-app advertisements get in the way of the graphs, and the graphs are fairly small with no way of changing the scale of either the horizontal or vertical axes.

CPU Monitor is also able to generate a list of applications that it deems most impacting on performance, based on their CPU usage.

Overall, the visualisation methods in CPU monitor are functional, but because their usability leaves some things to be desired, other profilers are better for generating insight in the usage of resources by the system.

5.4 General information

This section summarises some statistics on the number of downloads these applications have and their availability for download. It also lists the availability of additional resources, such as articles, whitepapers, user manuals and help files. The download information is copied from the Google Play store, which does not give very precise insight in the amount of downloads applications have. The information is summarised in Table 4.

5.4.1 Intel Performance Viewer

Intel Performance Viewer is available for download from the Google Play store. It has between 5 000 and 10 000 downloads. No other resources could be found on Intel's website or anywhere else.

5.4.2 PowerTutor

PowerTutor is available for download from the Google Play store. It has between 100 000 and 500 000 downloads.

The method used by PowerTutor to generate a power model of the device it is running on can be found in Accurate Online Power Estimation and Automatic Battery Behavior Based Power Model Generation for Smartphones (Zhang, 2010) [17].

Höpfner and Schirmer (2012) [3] compare the accuracy of PowerTutor's profiling results to baseline hardware measurements, and concludes that the deviation between the results of these measurement methods is 0,81%.

Apart from these results in the literature, PowerTutor also has a website [10] which contains a user manual, documentation, and a link to the source code.

5.4.3 eDoctor

Currently, eDoctor is unavailable for download. Xiao et al (2013) [8] declare in their paper that they were planning on releasing eDoctor in the Google Play store, but this has not yet happened. All functionalities and features ascribed to eDoctor in this paper are extracted from Xiao's work.

5.4.4 Trepn

Trepn is not available for download through Google Play, though it is downloadable from Qualcomm's website [4]. Users will have to register a Qualcomm developer account, download the `.apk` installation package, and manually install it on their device. Qualcomm has not published the number of downloads.

Qualcomm offers a user manual, installation manual and list of release notes on its website. It also offers a link to a plug-in for the Eclipse Integrated Development Environment, which is able to import profiling data directly into Eclipse for further analysis. Qualcomm also maintains a forum on which users can communicate errors and discuss possible improvements.

5.4.5 GSam Battery Monitor

GSam is available for download from the Google Play Store, and currently has between 500 000 and 1 million downloads.

GSam offers a link to a online user guide from within the profiler. This user guide is posted on the GSam Battery Monitor blog [7].

5.4.6 CPU Monitor

CPU Monitor is available for download from the Google Play Store, and currently has between 10 000 and 50 000 downloads.

CPU Monitor offers a link to a online help file from within the profiler. This online help file references CPU Logger (or CL for short), which is assumed to be the old name of this profiler.

6. COMPARISON

In the previous section, we have outlined the functionalities and other qualities of all profilers under consideration. We have grouped these functionalities in 4 categories. In this section we will compare them, based on the functionalities that they offer.

Since eDoctor is unavailable for download, it will not be included in the comparison because we were unable to test the functionalities firsthand.

6.1 High-level functionality

High-level functionality was previously defined as the overall tasks that the profiler can perform.

6.1.1 Per component/systemwide

Most profilers are able to measure the use of hardware components over the whole device. Of these profilers, PowerTutor and Trepn offer the most useful information for developers.

PowerTutor is able to generate a very detailed overview of the amount of energy that each individual component uses. It is also able to export this data in a plain text format, which allows for detailed analysis.

PowerTutor is only able to calculate the energy usage per component. If a developer wants insight in the actual usage statistics for hardware components, Trepn is a better choice. Trepn can store this data in a format that it can read itself, or it can store it in a `.csv` format which can easily be imported in a spreadsheet program for further analysis.

6.1.2 Per application

Table 4. Availability of profilers

	Available for download?	#Downloads	Google Play?
Intel Performance Viewer	Yes	5K-10K	Yes
PowerTutor	Yes	100K-500K	Yes
eDoctor	No	-	No
Trepn	Yes	Unknown	No
GSam Battery Monitor	Yes	500K-1M	Yes
CPU Monitor	Yes	10K-50K	Yes

All profilers offer some functionality in this area. Of these profilers, PowerTutor is the best application to use for measuring the amount of energy used by an application. Intel Performance Viewer and Trepn offer overlay graphs that can offer some insight in resource usage as an application is being used.

6.1.3 Online profiling

All profilers except GSam Battery Monitor offer some functionality in this area. For online profiling of an application, it is useful to be able to see how CPU and memory usage, among other things, vary during use. Intel Performance Viewer and Trepn offer graphical overlays that enable developers to inspect the usage of these statistics while using the application.

CPU Monitor and Trepn offer comparable functionality when profiling the entire device. However, since Trepn's graphs are more easily readable and scalable, it surpasses CPU Monitor in this respect.

6.2 Component-level functionality

In this section, we will compare the functionality that profilers offer based on their ability to profile hardware components. Except for GSam Battery Monitor, the profilers offer a very similar set of hardware components that they can profile.

PowerTutor is the only application that can accurately measure the amount of energy that is used by each component. It can measure this energy usage system-wide, or decompose it per application. For developers that are interested in analysing what components are responsible for the high battery footprint of their application, PowerTutor is the best option.

Trepn is able to generate graphs in which a developer can see how the usage of hardware components varies over time. Intel Performance Viewer uses an overlay graph that developers can use to get a quick read on the behaviour of their application. Using either one of these profilers can help developers gain insight in how their application uses resources.

6.3 Data visualisation

In this section, we will compare the data visualisation methods that applications offer. These are mostly in the form of line graphs, displaying the usage of some resource over time.

The only visualisation Intel Performance Viewer offers is an overlay with line graphs for each of the data points that the user wants to track. This overlay is functional and offers a good insight in the real-time usage of resources of an application. The overlay graph of Intel Performance Viewer is slightly obtrusive in that it may make part of the screen harder to read, but it does not block touch.

Trepn offers similar functionality, although its overlays get in the way of the underlying application. Trepn's widgets are more customisable and offer a clearer insight in the real-time usage components by the system. Additionally, Trepn is able to store the information after profiling has been completed.

PowerTutor can generate pie charts that show on what components applications use energy. Additionally, it can create a pie chart that shows what portion of the total amount of energy used by the device is spent on each component.

PowerTutor, Trepn and CPU Monitor are all able to display component usage information about the entire device in line graphs. They are fairly similar in terms of the number of data points they trace. Trepn's visualisation is most useful, since users can choose whether they want to display the information in one combined line graph or in separate views. In both views, the user can scale the graphs. With PowerTutor and CPU Monitor, users are unable to do this.

7. CONCLUSION

This paper compared energy profilers for Android smartphones. We conducted a thorough search and made a selection of 6 energy profilers. We then scored these profilers on what functionalities they offer, and grouped these in meaningful categories. After this, we compared the different profilers on these functionalities.

For the purpose of measuring the amount of energy that applications use, PowerTutor is the best choice.

For measuring the actual usage of components, Trepn is a good choice. It is able to analyse the usage of some components that other profilers are unable to profile (notably GPU and GPS), and offers the best visualisation of their usage.

If a developer wants to get a quick insight in the resource usage of an application, Intel Performance Viewer performs well. The main advantages of Intel Performance Viewer over Trepn are that it is easier to download and install (since Trepn is not available in the Google Play Store) and the overlay graph is less obtrusive.

CPU Monitor offers a feature set that might be of interest to developers, but is outshadowed in all areas by other profilers.

Since the perceived quality of a profiler is subjective, the results in this paper can also appear subjective. There was not one profiler that was clearly better than all others. The reason for this is that, while PowerTutor is a clear winner in determining the energy usage of applications, Trepn is better at tracking the actual usage of components, and Intel Performance Viewer is better at real-time profiling.

In this research we focused mainly on the functionalities that profilers offer. Usability is also an important aspect in assessing the quality of a profiler, however during the research phase there was no noticeable difference on this subject between different profilers. The accuracy of the measurements of energy profilers is also an important aspect. However, comparison on this criterion would give an

incomplete image, as the profilers under consideration do not overlap strongly on what it is they measure.

8. FUTURE WORK

Since this study compares profilers available today, in the state that they are today, it will get outdated over time as new profilers become available and existing profilers receive updates. Future work would consist of a search for new profilers and a comparison against the ones in this paper.

This study focused mainly on the functionalities of profilers, and the components they can analyse. Since there were no serious usability issues that prevented the profilers from being used properly, usability was not assessed. Future work could comprise of a usability study of different profilers.

9. REFERENCES

[1] Fangwei Ding, Feng Xia, Wei Zhang, Xuhai Zhao, and Chengchuan Ma. Monitoring energy consumption of smartphones. In *Internet of Things (iThings/CPSCom), 2011 International Conference on and 4th International Conference on Cyber, Physical and Social Computing*, pages 610–613. IEEE, 2011.

[2] Mian Dong and Lin Zhong. Self-constructive high-rate system energy modeling for battery-powered mobile systems. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*, pages 335–348. ACM, 2011.

[3] Hagen Höpfner, Maximilian Schirmer, and Christian Bunse. On measuring smartphones' software energy requirements. In *ICSOFT*, pages 165–171, 2012.

[4] Qualcomm Inc. Trepn profiler, February 2013. From developer.qualcomm.com, accessed April 2014.

[5] Jar. Peak battery: Why smartphone battery life still stinks, and will for years. From techland.time.com, accessed April 2014.

[6] Wonwoo Jung, Chulkoo Kang, Chanmin Yoon, Donwon Kim, and Hojung Cha. Devscope: a nonintrusive and online power analysis tool for smartphone hardware components. In *Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis*, pages 353–362. ACM, 2012.

[7] GSam Labs. Frequently Asked Questions - GSam Battery Monitor, January 2013. From badassbaterymonitor.blogspot.com, accessed May 2014.

[8] Xiao Ma, Peng Huang, Xinxin Jin, Pei Wang, Soyeon Park, Dongcai Shen, Yuanyuan Zhou, Lawrence K Saul, and Geoffrey M Voelker. edoctor: Automatically diagnosing abnormal battery drain issues on smartphones. In *Presented as part of the 10th USENIX Symposium on Networked Systems Design and Implementation*, pages 57–70. USENIX, 2013.

[9] Yeayeun Park, Mark Serrano, and Craig Pentrack. Battrace: Android battery performance testing via system call tracing.

[10] PowerTutor. A power monitor for android-based mobile platforms, October 2011.

[11] Google Play Store. Battery log, play.google.com. Accessed April 2014.

[12] Google Play Store. Cpu monitor, play.google.com. Accessed April 2014.

[13] Google Play Store. Dr power, play.google.com. Accessed April 2014.

[14] Google Play Store. Gsam battery monitor, play.google.com. Accessed April 2014.

[15] Google Play Store. Intel performance viewer, play.google.com. Accessed April 2014.

[16] Chanmin Yoon, Dongwon Kim, Wonwoo Jung, Chulkoo Kang, and Hojung Cha. Appscope: Application energy metering framework for android smartphone using kernel activity monitoring. In *USENIX ATC*, 2012.

[17] Lide Zhang, Birjodh Tiwana, Zhiyun Qian, Zhaoguang Wang, Robert P Dick, Zhuoqing Morley Mao, and Lei Yang. Accurate online power estimation and automatic battery behavior based power model generation for smartphones. In *Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis*, pages 105–114. ACM, 2010.