

[0004] Over recent years, the popularity of smartphones has dramatically increased. This has led to a widespread availability of smartphone applications. Further, smartphones have improved exponentially in terms of processing speed and memory capacity. This improvement has allowed application developers to create increasingly complex applications for such devices. Also, modern smartphones are equipped with a wide range of sensors and input/output (I/O) components such as GPS and WiFi receivers as well as cameras. These I/O components allow developers to create a diverse set of applications. In spite of such high computation power and developer flexibility, the battery-life of these modern smartphones has improved only two or three times over the last decade or so. Since smartphones operate on a limited amount of battery power, it is important to develop tools and techniques that aid in power-efficient application development.

[0005] Thus, what is needed is a software productivity tool suitable for detecting energy inefficiencies and capable of detecting energy bugs and energy hotspots. Furthermore, other desirable features and characteristics will become apparent from the subsequent detailed description and the appended claims, taken in conjunction with the accompanying drawings and this background of the disclosure.

SUMMARY

[0006] Embodiments of the present invention provide a software productivity tool comprising a detection module and a guidance module.

[0007] According to a first aspect, there is provided a software productivity tool comprising: a detection module adapted to automatically detect a power inefficiency in a device comprising hardware components and to calculate an E/U ratio indicating power consumption to utilization of hardware components over a time

period, the power consumption and the utilization of the hardware components being caused by a software program, the E/U ratio used for detection of the power inefficiency in a device on which the software program is run; and a guidance module adapted to automatically generate at least some of a plurality of test inputs receivable by the software program and determine which of the generated test inputs will cause the power inefficiency in the device on which the software program is run.

[0008] According to a second aspect, there is provided a software productivity tool comprising: a detection module adapted to automatically detect a power inefficiency in a device comprising hardware components and to detect a presence of at least one event sequence likely to cause a power inefficiency in the device on which a software program is run, the at least one event sequence being triggered by at least one input receivable by the software program; and a guidance module adapted to automatically generate some of a plurality of test inputs receivable by the software program and determine which of the generated test inputs cause a power inefficiency in the device, the plurality of test inputs comprising the at least one input adapted to trigger the at least one event sequence.

[0009] According to a third aspect, there is provided a system for identification of power inefficiencies, the system comprising: a portable device having an energy storage device for supplying power thereto, the portable device being configured to run software programs thereon; and a software productivity tool coupled to the portable device for identifying power inefficiencies in operation of one or more of the software programs by the portable device, the software productivity tool comprising: a detection module adapted to automatically detect a power inefficiency in a device comprising hardware components and to detect a presence of an event sequence likely to cause a power inefficiency in the portable device on which the one or more of the software programs are run, the event sequence being triggered

by an input receivable by the one or more of the software programs; and a guidance module adapted to automatically generate some of a plurality of test inputs receivable by the one or more of the software programs and determine which of the generated test inputs are likely to cause the power inefficiencies in the portable device, the plurality of test inputs comprising the at least one input adapted to trigger the event sequence.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] The accompanying figures, where like reference numerals refer to identical or functionally similar elements throughout the separate views and which together with the detailed description below are incorporated in and form part of the specification, serve to illustrate various embodiments and to explain various principles and advantages in accordance with a present invention, by way of non-limiting example only, wherein:

[0011] FIG. 1 depicts a table of a classification of exemplary energy bugs and energy hotspots in accordance with a present embodiment.

[0012] FIG. 2 depicts a diagram of an exemplary lifecycle of a software program activity on a device in accordance with the present embodiment.

[0013] FIG. 3 depicts a snippet of application code of an exemplary software program with a potential energy bug in accordance with the present embodiment.

[0014] FIG. 4, comprising FIGs. 4A and 4B, depicts snippets of application code of exemplary software programs in accordance with the present embodiment, wherein FIG. 4A depicts a snippet of application code comprising an energy hotspot due to disaggregated communication and FIG. 4B depicts the snippet of application code without the energy hotspot.

[0015] FIG. 5 depicts a bar graph of a power profile for a smartphone portable device in accordance with the present embodiment.

CLAIMS

1. A software productivity tool comprising:
a detection module adapted to automatically detect a power inefficiency in a device comprising hardware components and to calculate an E/U ratio indicating power consumption to utilization of hardware components over a time period, the power consumption and the utilization of the hardware components being caused by a software program, the E/U ratio used for detection of the power inefficiency in a device on which the software program is run; and
a guidance module adapted to automatically generate at least some of a plurality of test inputs receivable by the software program and determine which of the generated test inputs will cause the power inefficiency in the device on which the software program is run.
2. The software productivity tool in accordance with Claim 1 wherein the guidance module is further adapted to extract features from the at least some of the plurality of test inputs so as to determine which of the generated test inputs will cause a power inefficiency in the device.
3. The software productivity tool in accordance with Claim 2 wherein the extracted features are used to generate a model, the model being adapted to detect test inputs which cause power inefficiencies in the software program.
4. The software productivity tool in accordance with Claim 1 wherein the guidance module ranks determines which of the generated test inputs will cause a power inefficiency in the device in response to event sequences resulting from the software program running the generated test inputs, and wherein the guidance module ranks the event sequences in response to one or more of number of system calls in the event sequence, similarity to previously identified power inefficient event sequences, and starvation of the event sequence due to unexplored system calls in the event sequence.
5. The software productivity tool in accordance with any of the proceeding claims further comprising a detection module adapted to detect a presence of at least one event sequence based on the generated model, the at least one event sequence being triggered by at least one of the plurality of test inputs and likely to cause a power inefficiency in the device.

6. The software productivity tool in accordance with any of the proceeding claims wherein the software productivity tool is adapted for use on an Android platform.
7. A software productivity tool comprising:
 - a detection module adapted to automatically detect a power inefficiency in a device comprising hardware components and to detect a presence of at least one event sequence likely to cause a power inefficiency in the device on which a software program is run, the at least one event sequence being triggered by at least one input receivable by the software program; and
 - a guidance module adapted to automatically generate some of a plurality of test inputs receivable by the software program and determine which of the generated test inputs cause a power inefficiency in the device, the plurality of test inputs comprising the at least one input adapted to trigger the at least one event sequence.
8. The software productivity tool in accordance with Claim 7 wherein the presence of the at least one event sequence is detected based on the determination of the generated test inputs.
9. The software productivity tool in accordance with Claim 7 or 8 wherein the detection module is further adapted to calculate an E/U ratio used for detecting power consumption to utilization of hardware components over a time period, the device comprising the hardware components and the power consumption and the utilization of the hardware components being caused by the software program.
10. The software productivity tool in accordance with Claim 9 wherein the E/U ratio is calculated during a period before executing the software program, a period during executing the software program, and a period after executing the software program.
11. The software productivity tool in accordance with Claim 9 wherein the detection module is further adapted to detect an energy bug by comparing the E/U ratio calculated in the period before executing the software program to the E/U ratio calculated in the period after executing the software program using a statistical method, the energy bug being the at least one event sequence during which executing the software program prevents the device from being idle.

12. The software productivity tool in accordance with Claim 9 wherein the detection module is further adapted to detect an energy hotspot by analyzing the E/U ratio calculated during the period of executing the software program using an anomaly detection technique, the energy hotspot being the at least one event sequence during which executing the software program causes the device to consume power higher than required.

13. The software productivity tool in accordance with any of Claims 9 to 12 wherein the detection module is further adapted to receive a reading from a power meter attached to the device to determine the power consumption to calculate the E/U ratio.

14. The software productivity tool in accordance with any of Claims 9 to 13 wherein the software productivity tool is adapted for use on an Android platform and wherein the detection module is further adapted to receive a reading from an Android debug bridge attached to the device to determine the utilization of the hardware components of the device to calculate the E/U ratio.

15. The software productivity tool in accordance with Claim 1 wherein the E/U ratio is calculated during a period before executing the software program, a period during executing the software program, and a period after executing the software program.

16. The software productivity tool in accordance with Claim 15 wherein the detection module is further adapted to detect an energy bug by comparing the E/U ratio calculated in the period before executing the software program to the E/U ratio calculated in the period after executing the software program using a statistical method, the energy bug being the at least one event sequence during which executing the software program prevents the device from being idle.

17. The software productivity tool in accordance with Claim 15 wherein the detection module is further adapted to detect an energy hotspot by analyzing the E/U ratio calculated during the period of executing the software program using an anomaly detection technique, the energy hotspot being the at least one event sequence during which executing the software program causes the device to consume power higher than required.

18. The software productivity tool in accordance with any of Claims 1, 15 to 17 wherein the software productivity tool is adapted for use on an Android platform, and wherein the device is attached to a power meter and an Android debug bridge, and wherein the detection module is further adapted to receive a reading from the power meter to determine the power consumption to calculate the E/U ratio and receive a reading from the Android debug bridge to determine the utilization of the hardware components of the device to calculate the E/U ratio.

19. A system for identification of power inefficiencies, the system comprising:
a portable device having an energy storage device for supplying power thereto, the portable device being configured to run software programs thereon; and
a software productivity tool coupled to the portable device for identifying power inefficiencies in operation of one or more of the software programs by the portable device, the software productivity tool comprising:
a detection module adapted to automatically detect a power inefficiency in a device comprising hardware components and to detect a presence of an event sequence likely to cause a power inefficiency in the portable device on which the one or more of the software programs are run, the event sequence being triggered by an input receivable by the one or more of the software programs; and
a guidance module adapted to automatically generate some of a plurality of test inputs receivable by the one or more of the software programs and determine which of the generated test inputs are likely to cause the power inefficiencies in the portable device, the plurality of test inputs comprising the at least one input adapted to trigger the event sequence.

20. The system in accordance with Claim 19 wherein the portable device operates on an Android platform.

21. The system in accordance with Claim 20 further comprising:
a power meter coupled to the portable device for obtaining energy consumption data therefrom; and
an Android debug bridge coupled to the portable device for obtaining hardware component utilization data therefrom corresponding to utilization of hardware components of the portable device,

wherein the detection module is coupled to the power meter and the Android debug bridge for calculating an E/U ratio used for detecting power consumption to utilization of hardware components by the software program over a time period in response to the energy consumption data and the hardware component utilization data.

22. The system in accordance with Claim 21 wherein the E/U ratio is calculated during a period before executing the software program, a period during executing the software program, and a period after executing the software program, and wherein the detection module is further adapted to determine whether the event sequence is an energy bug which prevents the portable device from being idle and/or an energy hotspot which causes the portable device to consume power higher than required, the detection module detecting the energy bug by comparing the E/U ratio calculated in the period before executing the software program to the E/U ratio calculated in the period after executing the software program using a statistical method and detecting the energy hotspot by analyzing the E/U ratio calculated during the period of executing the software program using an anomaly detection technique.

23. The system in accordance with Claim 22 wherein the guidance module ranks the event sequence in response to one or more of a number of system calls in the event sequence, a similarity of the event sequence to previously identified energy bug and/or energy hotspot event sequences, and starvation of the event sequence due to unexplored system calls.