Proceedings of the International Conference on
Information Technologies (InfoTech-2018)
20-21 September 2018, Bulgaria

INFORMATION TECHNOLOGY FOR GREEN SOFTWARE ENGINEERING FOR THE INTERNET OF THINGS

Elena Stetsuyk, Dmitry Maevsky, Elena Maevskaya

Odessa National Polytechnic University
e-mail: Dmitry.A.Maevsky@opu.ua
Ukraine

Abstract: This article describes an information technology for green software engineering for IoT devices. For the using of proposed information technology, we should know only source assembler code of the program and the types of CPU and RAM. This technology consists of three steps: scanning of assembler code and determination types of each assembler instruction and their volume in bits; applying of the method for computer's estimating power consumption based on assembler source code and finally estimating of RAM energy consumption. Proposed information technology especially useful for development of software for the devices of Internet of Things.
Key words: Green Software, Energy Consumption, Software engineering, RAM power consumption, IoT programming.

1. INTRODUCTION

There are many software engineering technologies used in software development practice. All these technologies are aimed at the rapid and effective development of the structure of software and its code. In general, software development technology is a system of engineering principles for creating cost-effective software with specified quality characteristics [1]. Software engineering technologies include a same kind of steps [2] and differs only of its sequence. The sequences of steps depend from chosen model, such as Waterfall model [3, 4], Spiral model [5, 6], Chaos model [7] and many others. The main aim of listed above technologies is increasing of software quality by which is meant usability and dependability.

This quality requirements are very important for the all kinds of software. But there is another class of software by which meet this requirement is not enough.

Software for Internet of Things (IoT) devices was meant. Apart from the usability and dependability this software should have another requirement – energy efficiency.

Security and privacy are the biggest issues for IoT. All devices and collecting systems hold a lot of personal data about people - a smart meter knows when you are at home and who uses electronics when you are there - and this is shared with other devices and is contained in the databases of companies. With billions of devices connected together, people need to be sure that their information remains secure. Will someone be able to break the toaster and thus gain access to your entire network. IoT also opens up even more security threats to companies around the world. Safety experts say that doing is not enough to enhance the security and privacy of the IoT at the current early stages and, to prove its point, has broken a lot of means from a connected video player to automated lighting and intelligent refrigerators, as well as citywide systems such as traffic lights.

Another words, software for IoT should be green. Why it is so important? Let us look at the statistics.

All IoT devices work on computers (most popular - Raspberry Pi). The latest Raspberry Pi3 model consumes 3.6 watts [8]. IoT is gaining in popularity every year, and by 2020, about 38.73 billion devices are forecast.

A smartphone can be considered the most popular device in our time. The easiest smartphone consumes about 7 watts [9]. By 2020, the number of smartphones is forecast to increase, namely 2.87 billion.

A large number of laptop users are being tracked around the world. The average power consumption of laptops per hour is 60 watts [10]. By 2020, projected growth of laptops - up to 1545 million.

The number of desktop computers is growing every day and by 2020 their number will be about 200 million devices. On average, a personal computer consumes 200 watts [11].

It is important to note the growth in the number of tablets in the coming years. It is projected that by 2020, tablets will be larger than laptops, namely 185 million. On average, tablets consume 20 watts [12].

Table 1 shows the amount of equipment by 2020, the power of each unit of the device and the total power of each group of devices.

Table 1. Number of devices and their power consumption

	Unit power, watts	*Quantity, billion*	*Power of all devices, MW*
Desktop	*200*	*0,2*	*40000*
Laptop	*60*	*0,32*	*19200*
Tablet	*20*	*1,46*	*29200*
Smartphone	*7*	*6,1*	*42700*
IoT	*3.6*	*30*	*108000*
Total:			*239100*

By summing up the power consumption of all devices, by 2020, the total energy consumption of devices around the world will be near 240000 MW. For comparison, the data of the largest power plant in the world, called Tu Tuoketuo, located in China. Tuoketuo is the largest station in the world. Installed capacity is 6600 MW.

2. HOW THE SOFTWARE CAN BECAME GREEN?

The IoT software can be considered green if it helps save resources. The program is information, that is, an immaterial object that does not directly affect the objects of our material world. All such hardware can be conventionally divided into two groups: devices that directly provide software operation, and devices controlled by this software [13]. The first group includes the hardware of the computer, and the second - the various peripheral devices connected to it. Therefore, green software should help save resources that are consumed both directly by the computer and its peripheral devices.

The software does not always function as intended by the developers. One of the reasons for such a malfunction is the defects made in the program at the stage of its creation, modification, or correction of other defects. The process of detection of these defects in time is the subject of study of a separate scientific direction - the reliability of software. In the case of software critical systems, the correct functioning of them is a major factor. Software defects lead to errors in its functioning, which in turn can lead to man-made accidents, environmental disasters, and human casualties. Thus, if we are talking about preserving resources, then reliability becomes a necessary requirement, which must necessarily be presented to green software.

We can conclude that the software, to become green, must meet the following requirements:

1. The requirement to minimize the resources required for functioning of software. These resources include the volume of operational computer memory, clock speed and CPU performance.
2. The requirement to save resources consumed by peripherals, which software controls.
3. The requirement for reliability of functioning and stability in relation to external target actions aimed at either termination or control of the IoT software.

This article is devoted to creation of information technology, focused on creation of green IoT software, which satisfies the first two requirements. The main feature of the models and methods implemented in this technology is the ability to perform power consumption estimation of computing devices based on the source code. This approach especially reasonable at the stage of software development, when from several algorithms, which identical in the performed functions, it is required to choose the most optimal algorithm from the point of view of power consumption.

The energy consumption is estimated only for RAM. There are several reasons for this.

First, RAM, along with the CPU, is an integral part of all computing devices, and in particular - devices of the IoT.

Secondly, RAM in modern computing devices is in second place in terms of power consumption after the CPU. So, according to [8], for a computer with a Xeon 5310 processor, RAM consumes 20% of all energy (the Xeon processor 5310 consumes 40%). Therefore, optimizing the program code from the point of view of the power consumption of RAM will significantly reduce the power consumption of the entire device.

And thirdly. Currently, the CPU has built-in tools for monitoring power consumption. RAM does not have such tools, so the task of developing information technology, which allows to include optimal algorithms at the stage of software development, is topical.

The next section describes the structure of information technology, which allows you to compare several versions of the program code by the criterion of the amount of energy consumed by the RAM when they are executed.

3. STRUCTURE OF INFORMATION TECHNOLOGY

The term "information technology" refers to "a set of techniques, manufacturing processes and software and hardware integrated with the aim of collecting, processing, storage, distribution, display and use of information for the benefit of its users". In fact, the definition is not on the one and the many processes that are based on certain scientific positions, models and methods. The initial material and the final product of these processes is information. Thus, we can say that the purpose of any information technology is to create an information product. Based on this interpretation, information technology can be represented as shown in Fig. 1.

The basis of information technology is its theoretical basis. On the created theoretical basis, the imaging models are based refer to the process of creating the information product of the real world. It is these processes and their interconnection that make up the domain of the information system that implements the technology. The set of models, as well as rules and methods of their application forms the methodological basis of information technology. To create an information product, you need the appropriate means (means of production). Today, special information systems, such as Java Eclipse, Microsoft Visual Studio, Delphi, and many others, serve as such means of production.

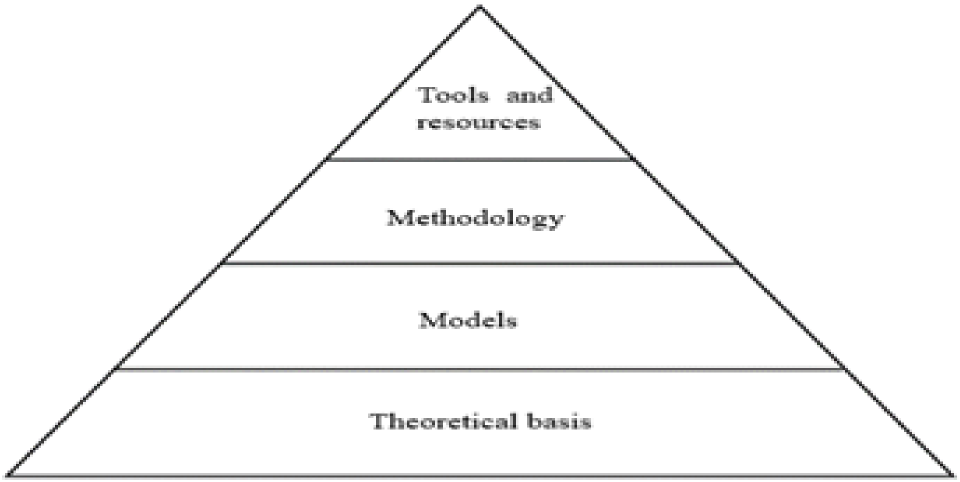

Fig. 1. Ingredients of information technology

In the process of IP use resources - information (input), energy, material, time and so on. The minimization of consumption of these resources is the goal of green technologies. Thus, we can conclude that speaking about green information technologies, we must understand not only the top of the pyramid - information systems, but also other components - the theory, models and methods. And the information technology itself can be considered green, only if all its components are green.

Below is a description of all the components of the information technology comparing several versions of the program code by the criterion of the amount of energy consumed by the RAM when they are executed.

3.1. Theoretical basis and models

Theoretical basis of information technology of comparing software code versions is formed by two mathematical models. The first mathematical model serves to determine the total power that is spent on transferring one bit of RAM from the logical zero state to the logical one state. This model is based on the calculation of the transient in the RAM cell and is described in detail in [14].

$$S = E \cdot \sqrt{\frac{1}{T} \cdot \left[\frac{A_1^2}{2p_1} \cdot \left(e^{2\alpha_1 T} - 1\right) + \frac{2A_1A_2}{p_1 + p_2} \cdot \left(e^{(\alpha_1+\alpha_2)T} - 1\right) + \frac{A_2^2}{2p_2} \cdot \left(e^{2\alpha_2 T} - 1\right) \right]}, \qquad (1)$$

where:

$$\alpha_1 = \frac{-\left(R_1C_1 + R_2C_2 + R_2C_1\right) + \sqrt{\left(R_1C_1 + R_2C_2 + R_2C_1\right)^2 - 4 \cdot R_1R_2C_1C_2}}{2 \cdot R_1R_2C_1C_2},$$

$$\alpha_2 = \frac{-\left(R_1C_1 + R_2C_2 + R_2C_1\right) - \sqrt{\left(R_1C_1 + R_2C_2 + R_2C_1\right)^2 - 4 \cdot R_1R_2C_1C_2}}{2 \cdot R_1R_2C_1C_2}$$

$$A_1 = -\frac{\left(p_2 \cdot R_1R_2C_1C_2 + R_1C_1\right)\cdot\left(V_{1H} - V_{REF}\right) + R_2C_1\left(V_{REF} - V_{1L}\right)}{R_2 \cdot \sqrt{\left(R_1C_1 + R_2C_2 + R_2C_1\right)^2 - 4\cdot R_1R_2C_1C_2}},$$

$$A_2 = \frac{\left(p_1 \cdot R_1R_2C_1C_2 + R_1C_1\right)\cdot\left(V_{1H} - V_{REF}\right) + R_2C_1\left(V_{REF} - V_{1L}\right)}{R_2 \cdot \sqrt{\left(R_1C_1 + R_2C_2 + R_2C_1\right)^2 - 4\cdot R_1R_2C_1C_2}}$$

In these formulas: V_{IH} – logical "1" voltage; V_{REF} – voltage in the information bus; V_{IL} – logical "0" voltage; R_1 – information bus resistance; R_2 – Source-Drain resistance of field-effect transistor when it turns on; C_1 – information bus capacitance; C_2 – capacitance of the capacitor in Basic DRAM Cell. These parameters can be found in the technical documentation for RAM, for example, in [15].

This mathematical model allows you to determine the power consumed by changing the state from "0" to "1" of only one bit of RAM. However, when executing program code, not one bit is changed, but much more. The exact number depends not only on the bit depth of the operation, but also on the information that is being processed at the moment. It is impossible to predict this exact number. We can calculate only the mathematical expectation of this quantity. The second mathematical model serves to calculate the mathematical expectation of the number of bits that pass from the logical "0" state to the logical "1" state. All stages of its construction are considered in detail in [9]. In this paper we give only a finite formula:

$$M(r) = -0.00011791\cdot r^3 + 0.00444567\cdot r^2 + 0.19515112\cdot r + 0.22247721. \qquad (2)$$

Here: M(r) – desired mathematical expectation, and r – capacity of operation.

3.2. Methodology

Based on the mathematical models described in 3.1, a method has been developed for determining the energy consumed by the operating memory when executing the program code. This method consists of the following steps:

1. To get Assembler code of the researched program.
2. To carry out the analysis of each of the instructions of Assembler code and determine the number of clock signals, which are necessary for the instruction execution and RAM area length, which is changed by this instruction. The number of clock signals can be learnt in CPU specifications, but the RAM area length corresponds to instruction operands length.
3. To calculate the total number of CPU clock signals necessary for a single program execution on the basis of analysis described in previous paragraph. We use Nt to denote the calculated number of clock signals.
4. To calculate the expected value of the bit amount changed by this instruction on the basis of RAM area length changed by each of the instructions and with the help of mathematical model developed in paragraph 5.
5. To calculate the complex power spent in order to change the RAM state by the instruction with number k according to the formula:

$$S_{rk} = M\left(r_k\right)\cdot S_1,$$

where: r – RAM area length changed by the instruction (bit capacity of the operation); M(r) – expected value of the number of changed from “0” to “1” bits depending on the RAM area length calculated according to the formula (2); S_1 – the complex power used in order to change 1 RAM bit state according to model (1).

6. To calculate the complex power used in order to change all RAM area states during a single program execution as a sum of complex powers S_k for every separate instruction. We use S_Σ to denote this complex power:

$$S_\Sigma = \sum_{k=1}^{N} S_{rk} .$$

7. To calculate a specific power W by the computer for a one second of program execution according to the formula:

$$W = \frac{N_t}{f_t} \cdot S_\Sigma ,$$

where: W – consumed energy, Nt – the number of CPU clock signals necessary for a single program execution, ft – frequency of CPU clock signal.

Value W is a final result of the energy consumption evaluation method in executing the programs on the basis of their source code.

3.3. Program Tool

Based on the models described above, information technology was developed that will help determine the program's energy consumption based on its source code. It is easier to present an information system in the form of a JDEF0 diagram. controls and mechanisms that are detailed (decomposed) to the required level by inputs and outputs. Using IDEF0-diagrams it is possible to easily define and describe the important processes of any technology to show their correct sequence. IDEF0-diagram is very convenient to use, it is an intuitive methodology of functional modeling.

The IDEF0 designation consists of blocks, each of which is a "black box" with controls and mechanisms that are detailed (decomposed) to the required level by inputs and outputs. Based on IDEF0-diagrams, you can quickly identify and describe the key processes of any technology to show their correct sequence. IDEF0 is a very simple and at the same time intuitive methodology of functional modeling.

With this methodology, information can be transferred between developers, consultants and users. The methodology was very carefully developed, it is universal and convenient for designing. Below is the IDEF0 diagram (Figure 2) of the information technology for determining the energy consumed by the program code.

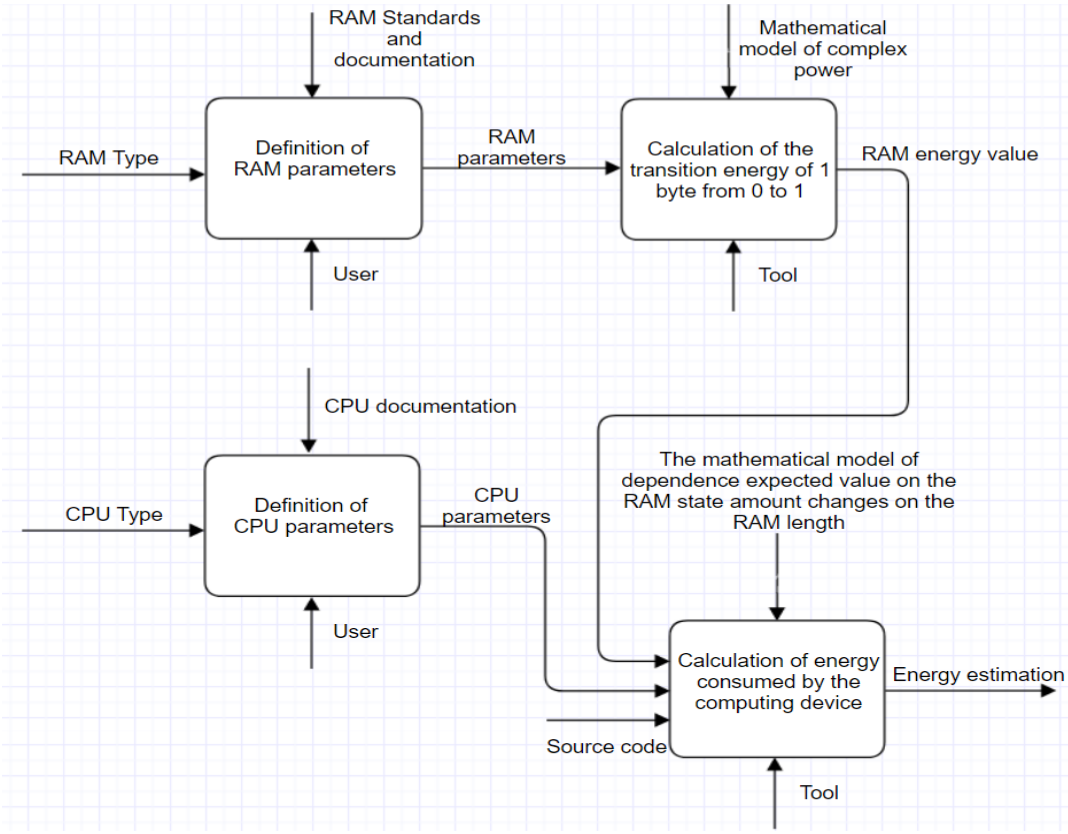

Fig. 2. IDEF0 diagram of the information technology for evaluating the IoT energy consumption

The program tool "ESTimation Energy Tool" (ESTET) includes an interface part that allows the user to select the required file with the code of the program that is being explored. The program includes a logical part in which all mathematical calculations are performed to determine the energy consumption. You can also select the type of processor and RAM of the computer on which the program is running and see the result of the calculations.

The main class of the created program tool is the *MainWindow.java* class. This class organizes the interconnection of all components of the program. Let's consider main methods of this class. CompParamWindow.java is a class that is used to work with lists of CPUs and types of RAM. NewMemoryWindow.java is the class that is required to add new RAM to the shared list. NewProcWindow.java is the class that is required to add new CPU to the shared list. AsmFileWindow.java is a method implemented to work with Assembler code. Logic.java is a class that is necessary for creating program logic, namely for creating algorithms and programming mathematical formulas. ResultTableWindow.java - this class is needed to work with the results of calculations.

4. CONCLUSION

This article describes the methodological foundations of information technology, which allows you to compare different versions of the program code of the different IoT devices by the criterion of power, which is consumed by the operating memory of the device. The proposed technology has merits and demerits, like everything that is created by human. Let's start with the merits.

The first advantage is that to evaluate the power you do not need to know anything except the source code and the electrical parameters of the RAM. These data are generally known.

The second advantage is the possibility of using the technology in the early stages of creating software products. At these stages, a choice is made of one or another algorithm that must implement one of the requirements for the functioning of the program. The proposed technology allows choosing among the several algorithms the best by the criterion of energy consumption. Thus, applying this technology, we get the opportunity to create software that is really green.

Now let's talk about the shortcomings of the proposed approach.

The first drawback is its approximate nature. The method of calculating the power consumption, which is applied in the proposed technology, is based on the calculation of the transient in the replacement circuit of the memory cell. However, this substitution scheme is not accurate and only approximately describes the electrical processes occurring in the operative memory. However, there is no other scheme for today.

The second drawback is the impossibility of taking into account the number of repetitions of individual cyclic sections of the program code. This defect is unavoidable and is a fundamental limitation. After all, the number of repetitions of loop operators depends on the source data. It is usually unknown at the stage of software development. To eliminate this limitation, the authors use a static code analysis while calculating the execution time of each program instruction. And then it is assumed that all the program code, as a single unit, is executed within one second. The number of repetitions of the code is determined by the total time that is necessary for a single execution of all instructions.

The third disadvantage relates to the developed software and consists in the fact that only the program code in the assembler language is analyzed.

The approach described in the article is only the first step towards creating methods for developing green software. Therefore, the elimination of the noted shortcomings is also the direction of its development.

REFERENCES

[1] N. Saher, F. Baharom, O. Ghazali **(2017)**. Requirement change taxonomy and categorization in agile software development. *Proceedings of the 2017 6th International Conference on Electrical Engineering and Informatics (ICEEI),* Langkawi, pp. 1-6, doi: 10.1109/ICEEI.2017.8312441

[2] E. A. Matusse, E. H. M. Huzita, T. F. C. Tait **(2012)**. Metrics and indicators to assist in the distribution of process steps for distributed software development: A systematic review. *Proceedings of the 2012 XXXVIII Conferencia Latinoamericana En Informatica (CLEI),* Medellin, pp. 1-10, doi: 10.1109/CLEI.2012.6427205

[3] R. Elghondakly, S. Moussa, N. Badr **(2015)**. Waterfall and agile requirements-based model for automated test cases generation," *Proceedings of 2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS)*, Cairo, pp. 607-612, doi: 10.1109/IntelCIS.2015.7397285

[4] P. Trivedi, A. Sharma **(2013)**. A comparative study between iterative waterfall and incremental software development life cycle model for optimizing the resources using computer simulation," *Proceedings of 2013 2nd International Conference on Information Management in the Knowledge Economy, Chandigarh*, pp. 188-194.

[5] G. Joshi, P. Desai **(2016)**. Building Software Testing Skills in Undergraduate Students Using Spiral Model Approach. *Proceedings of 2016 IEEE Eighth International Conference on Technology for Education (T4E), Mumbai*, pp. 244-245, doi: 10.1109/T4E.2016.061

[6] Richard W. Selby **(2007)**. Software Process: Early Spiral Model. In *Software Engineering: Barry W. Boehm's Lifetime Contributions to Software Development, Management, and Research*, Wiley-IEEE Press, doi: 10.1109/9780470187562.ch4

[7] L. Qian **(2014)**. Methodology on qualitative simulation modeling of software reliability based on chaos theory. *Proceedings of 2014 IEEE 5th International Conference on Software Engineering and Service Science*, Beijing, pp. 99-104, doi: 10.1109/ICSESS.2014.6933522

[8] Power Consumption Benchmarks (*available at:* https://www.pidramble.com/wiki/ benchmarks/power-consumption).

[9] Vernon Trollinger **(2016)**. Energy Efficiency, February (*available at:* http://www. bounceenergy.com/blog/2016/02/how-much-energy-does-this-appliance-use-phone-charger/).

[10] Matt Klein **(2016)**. How much energy does a regular laptop consume? (*available at:* https://www.quora.com/How-much-energy-does-a-regular-laptop-consume).

[11] Nicholas Brown **(2016).** How Much Power Do Computers Consume? **(***available at:* https://www.kompulsa.com/much-power-computers-consume/).

[12] Nick Vogt **(2016).** How Much Power Does a Phone or Tablet Use While Charging?
(*available at:* https://www.h3xed.com/mobile/how-much-power-does-a-phone-or-tablet-use-while-charging).

[13] Meyer, N., Januszewski, R. Research on energy-efficient computing solutions. PRACE Workshop “New Languages & FutureTechnology Prototypes” at Leibniz Supercompu-ting Centrein, Garching near Munich, 217-224 (2010).

[14] Maevsky, D. A., Maevskaya, E. J., Stetsuyk E. D. **(2017)**. Evaluating the RAM Energy Consumption at the Stage of Software Development. *Green IT Engineering: Concepts, Models, Complex Systems Architectures [Monongraph]*. – Springer International Publishing, p.p. 101 – 121, doi: 10.1007/978-3-319-44162-7_6

[15] Double Data Rate (DDR) SDRAM JESD79F. **(2008)**. In: JEDEC Standard. http://cs.ecs.baylor.edu/~maurer/CSI5338/JEDEC79R2.pdf