

# Model and Methods of Human-Centered Personal Computers Adaptive Power Control



Igor Turkin and Oleksandr Vdovitchenko

**Abstract** The problem of economical use of modern PCs' electric energy is solved at all levels, starting with hardware. Power consumption of the computing platform can be structured by dividing it into two parts: static and dynamic, where last of them is defined by processor load. There are two main approaches in solving the problem of reducing the computers' power consumption at the hardware and at the operating system levels:

- the dynamic change of the processor voltage and frequency;
- the advanced configuration's interface and power management technology.

Only a few publications that take into account the user's model when power saving PC's control are known. Mostly, the terms «user's model», «user's behavior model» are used in works on ergonomics of the user's interface, in search and advisory systems, in systems of user's authentication, and also in tasks of identifying abnormal behavior of computing systems. Proposed solutions use models based on the measurement of the user's motor activity with typical and most used devices for entering information: a keyboard and a manual manipulator «mouse». Also the existing models ignore the semantic and target components of the user's behavior for which the elementary motor activity is carried out. Such components are manifested at the level of the operating system in the form of application's starts/stops, switching of focus, etc., and they are also an integral part of the general model of user's activity of the PC. The purpose of the article is to develop a model of user's activity and the method of its identification in the system of PC's power consumption. In addition the evaluation of the implementation of this model and methods in the system of the PC's power consumption is of interest.

**Keywords** Adaptivity · The dynamic change of the condition · Hidden markov model · BAUM-Welch method · Energy savings

---

I. Turkin · O. Vdovitchenko (✉)  
National Aerospace University «KhAI», 17 Chkalova Street, Kharkiv, Ukraine  
e-mail: o.vdovitchenko@khai.edu

# 1 Introduction

## 1.1 Motivation

In computing systems dynamic power control technologies which are implemented at the software-hardware level provide great opportunities for dynamic control of productivity and consumed energy, but the control of these indicators is limited only by the use of resources and the factors of user's behavior that are not reflected in using of system resources such as processor load, time of inactivity, etc. are not always taken into account.

Such researchers as A. Chandrakasan, R. Brodersen, J. M. Rabaey, M. Pedram, S. Udani, J. Smith, L. Benini, G. De Michaels, J. Lorch, A. Smith, D. Snowdon made significant contribution to the theory of dynamic power consumption control of personal computers (PCs). The research is conducted at the level of international corporations, with Intel, Microsoft, Toshiba, Samsung, AMD, Asus as leaders.

There are few publications that take into account the user's model for energy-saving personal computer (PC) management. Mostly, the terms "user model", "user activity model" are used in works on ergonomics of the user interface, search and recommendation systems, user authentication systems, as well as in the tasks of identifying the abnormal behavior of computing systems, etc. Among the important scientific works, which use the user's model for energy-efficient PC control, let's mention publications by L. Benini, J.-B. Durand, B. Kveton, Chich-Han Yu, S. Mannor. The solutions which are proposed in these articles the models based on measuring the motor activity of a user with the typical and most usable devices for entering information—a keyboard and manual manipulator "mouse" are used. In the existing models, the semantic and target components of the user's activity for which the elementary motor activity is carried out, are ignored. These components are detected at the operating system level in the form of application starts/stops, focus switching, etc., and are also an inseparable part of the general model of PC user's activity, so developing a user's model and adaptive power control technology for personal computers is an important scientific and applied problem.

## 1.2 Goal and Approach

The purpose of the research is to reduce the energy costs of modern PCs, maintaining the acceptable for user characteristics of PC using comfort. To achieve this goal, the following tasks should be solved:

1. To do the analysis of publications concerning the problem of energy-efficient use of PCs;

2. To develop a user focused model of the computer system PC's adaptive power control;
3. To develop a user-centered adaptive power consumption control system for PCs;
4. To perform an experimental evaluation of the feasibility and effectiveness of the user model and the method of adaptive power control of PCs.

## 2 Related Work

In the era of information technologies, the problem of energy consumption has come to a new level. New technologies and standards are being developed. Companies that have joined the Climate Savers Computing Initiative have been working to reduce the worldwide energy consumption of computers by half in 2018. CSCI has already joined Apple, Dell, Google, HP, Intel, Lenovo, Microsoft and many others [1].

The power consumption of the computing platform can be structured by dividing it into two parts:

- the static part includes the costs of maintaining the platform in working state (for example, energy consumed by the cooler, motherboard, etc.). It has a weak correlation with the nature of executable applications, so we are less interested in it;
- the dynamic part is associated with the execution of programs. As this part directly depends on the characteristics of the loading, the degree of resources' utilization and the policy of power consumption, it can vary greatly. Exactly here there are opportunities for optimization, as the work of the software makes a significant contribution to the overall power consumption of the system [2].

It is known that there is no need for the constant use of the processor with maximum performance. When optimizing power consumption, we can consider the energy spent to perform all tasks in strictly defined time intervals as the criterion of efficiency. The constant work of the computer during idle time, even with reduced processor consumption, leads to excessive consumption of electricity.

Based on this, two energy-saving approaches were developed:

- the dynamic change of frequency and voltage of the processor's power supply;
- idle-waiting mode control.

SpeedStep is Intel's energy-saving technology, based on a dynamic change in the frequency and voltage of the processor. The development under the working name Geyserville includes modifications: SpeedStep, SpeedStep II and SpeedStep III [3, 4].

The similar energy-saving technologies from AMD are known. PowerNow! technology which is developed for application in processors' versions K6-2+, K6-III+ i Athlon, which are installed in laptops. The CPU clock frequency and

voltage automatically decrease when the computer is idle or load insufficiently, which reduces power consumption and heat dissipation. Cool'n'Quiet is a PowerNow! technology version designed for ordinary, non-mobile processors [5].

These energy saving technologies are based on the method of dynamic voltage and frequency scaling [6]. The supplied voltage determines the frequency of the processor, and, therefore, affects the power consumption. At the same time voltage switching is energy-intensive and rather long (expensive) transient processes. Accordingly, identifying the optimal frequency of the processor for performing the incoming tasks is a problem.

Advanced Configuration and Power Interface (ACPI) [7]—a common interface for hardware detection, power control and configuration of motherboard and devices. The task of ACPI is to provide interaction between the operating system, equipment and motherboard BIOS. ACPI was designed specifically for the implementation of OnNow's ideology. OnNow is an ideology of a computer device that is ready to recover and begin to work quickly at any time.

Models and methods of dynamic power control are based on policies. The purpose of the power control policy is turning off the required components when they are not in use and turning them on at the moment when they become necessary. The main problem with the power control policy is uncertainty at the time of the transition from the on state to the off state and vice versa. The problem of prognostication is as follows: it is necessary to determine whether the idle period will be long enough to compensate for the overhead of the transition between device states. The minimum length of the idle time for energy savings is called the break-even time [8]. Existing power control policies can be divided into three groups: time-out policies; predictive policies; stochastic policies.

Time-out policies are built upon measuring time of inactivity  $t$ . They put the device into sleep mode, if it is in standby mode, more than  $t$ . The main supposition is that if the device is not used during time  $t$ , then it will be put into idle state. The disadvantage of this policy is that devices may not be used while consuming energy until the end of a given time [9]. The solution to this problem is an adaptive time-out policy—a dynamic change  $t$  based on certain parameters. In the paper [10] regulation  $t$  based on the correlation between the performance delay and the waiting time for the previous idle period is proposed. If the coefficient is high,  $t$  will increase, otherwise—it decreases.

Prognostication policies determine the duration of idle time's future period. Thus, you can decide immediately about the transition to sleep, depending on the prediction. In the paper [11] the scheme of exponential averages is offered. The future period of idle time is predicted on the basis of measuring the exponential average provided and actual lengths of the previous period of idle time:

$$I_{n+1} = a \cdot J_n + (1 - a) \cdot I_n, \quad (1)$$

where

$I_{n+1}$  is a new predicted value of idle time;

- In the last predicted value;
- Jn the last period of idle time;
- a the constant damping factor in the range from 0 to 1

Chung and others in the paper [12] propose a prediction policy that uses adaptive tree learning. An adaptively trained tree encodes a sequence of idle time periods in the nodes of the tree. Sheet nodes preserve the level of reliability of prognostication (RDP) associated with the corresponding sequence. The RDP is updated with the help of a finite state machine.

A policy that represents a mixture of adaptive time-outs and prognostication schemes is considered in the paper [13]. The policy is based on separating the cluster sequence into groups which are referred as sessions, depending on user queries. It provides the length of the current session, based on the prognosticated and actual lengths of previous sessions. In the absence of requests from the user, the time-out time will be reduced by the adjustment factor instead of disconnecting the device after a specified time. Conversely, if a request is received, time is increased by the same adjustment factor. The off command is given when the device is in standby mode for a long time compared to the predicted session duration.

Stochastic policies are based on models of stochastic processes. When an accidental event occurs, the device changes the power state. Minimizing energy consumption and productivity delays are criteria for stochastic optimization.

In the paper [14] modeling the process of queries, queue queries and service provider (device) in the form of Markov processes with discrete time was carried out. They determine the state of the system, such as the concatenation of queuing states, queue, and service provider. The policy is a matrix, on the basis of which a team corresponding to the specific state of the system is selected. Policy optimization for maximizing stored energy while limiting productivity delays is implemented.

In the paper [15] service provider and query queues are modeled as Markov processes with continuous time, and the time between requests arrives is modeled as an exponential distribution. This model gives the opportunity not to evaluate the periodically necessary power supply. Instead, the arrival and maintenance of queries are events that initiate the transition between states.

In the papers [22, 23] the model of queries and their maintenance using «memoryless distributions», which is not accurate in real situations is proposed [16]. In the paper [17] the query arrival is modeled as Pareto distribution, and the times of device transitions between power states are modeled as a uniform distribution.

Existing power consumption policies have a number of disadvantages, for example:

- time-out policies continue to consume energy when performance is no longer needed, but the time to an end of the PC work has not come yet;
- prognostication policies can solve this problem, but they work on the fundamental assumption that service requests have a high degree of correlation;

- stochastic policies are good at minimizing power consumption, taking into account loss of productivity, but they assume an accurate distribution for the queue of queries, if the distribution changes, then the model will become inadequate.

### 3 Background of the Work

The basis for the article was the thesis «One design point that fits all use cases won't be sufficient to scale down the energy consumption to the required levels. Therefore, design and run-time tools supporting dynamic adaptation and adoption of energy under quality of experience requirements will be needed. The full complexity of dynamic adaptation should not percolate up to the application programming level but be part of the system level. This will require comprehensive energy models for computing.» expressed in the work «Green IT Engineering: Concepts, Models, Complex Systems Architectures» [18].

The concept of adaptability is dominant at the modern theory of energy-saving control of personal computer systems. In this paper, we touch upon a little researched area, exactly the user. He is regarded as the main influencing person in the power control system of a personal computer.

Analysis of user models in the task of managing PC's power consumption is one more essential point. The term «User model» is considered from different positions in modern information technologies. The user can be defined as a subject (action maker or state carrier), a simplified imaginary schema (subjective model) created by a user that is powerful enough to describe the interaction with a device, but does not always reflect its real internal mechanics [19]. The user as an object is understood as the one to whom the subject's thought or action is directed.

By purpose, user models as objects are divided into:

- adaptive interfaces. They are changing the display of information. They are built on the basis of queries (documents, words, history), human characteristics (demographic, individual, psychological) [20];
- information behavior models. They are changing the system reaction. They are built on the basis of human characteristics, parameters of specific tasks, i.e., the main characteristics of actions (time to solve the problem, time to act). They are composed of three levels: syntactic, semantic, pragmatic;
- model of identification user's rights.

By the method of obtaining a user models are divided into:

- template model is a set of criteria selected by experts which characterize the behavior of the user;
- dynamic model is a «user track» rating based on system reviews.

Turning to the management of PC's power consumption, you can identify the main parameters on which the control depends. In this interpretation the user model in the task of managing the power consumption of personal computers is considered in terms:

- User's waiting time is a value that characterizes the time that passed from the current state of the system in the working state;
- performance is the value that characterizes the performance of a personal computer, depending on the mode of power consumption;
- power consumption is a value that shows how much power your PC consumes.

The problem of the user model in the system of power consumption management of personal computers hasn't been studied well, so from the existing works it is possible to distinguish two approaches.

The first is the development of a user model based on the Bayesian network [21]. The main hypothesis is that the latent state significantly influences on the distribution of the work's duration. The future idle time is defined as the duration between the current state and the time of the next active state. The user model is described as the state in time with the parameters: the previous state, the time in the previous state, the peak load hours, the current state.

The second one is the user model, which represents the queue of requests for the service system. In the paper [22] it is assumed that the user can be in two states: active and passive. In the active state of the user there should be a queue of requests, which is exponentially distributed. The simulation of user interaction and energy control system is described as a Markov process with discrete time.

The disadvantage of existing models is that these models ignore the semantic and target components of user behavior for which elementary motor activity is carried out. Such component is manifested at the level of the operating system in the form of start/stop applications, switching focus, etc. and it is also an integral part of a model of PC's user activity. To evaluate the efficiency of this model, we use the energy saving characteristic proposed in the paper [23].

## 4 Proposed Solution for Adaptive Power Control of Pc

Model of human-centered personal computer adaptive power control system is presented below (Fig. 1). According to available classifications of adaptive systems, this system belongs to model identification adaptive controllers (MIACs), which perform system identification while the system is running.

Adaptability is realized by correcting the algorithm parameters of the decision-making block on controlling the power supply circuits (policies), as well as by using adapted (corrected) user activity models and PC power consumption processes. The interaction of a detailed stochastic model of a system with identification blocks and adaptive control is depicted in Fig. 2.

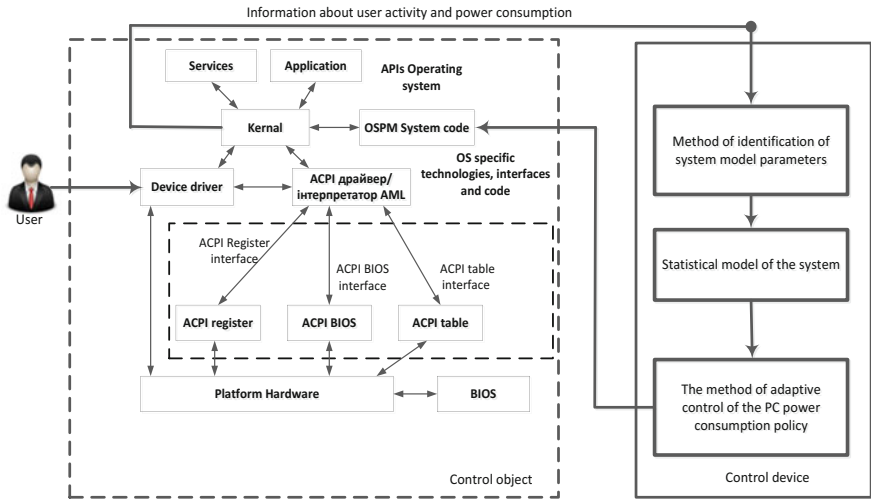


Fig. 1 Model of human-centered personal computer adaptive power control system

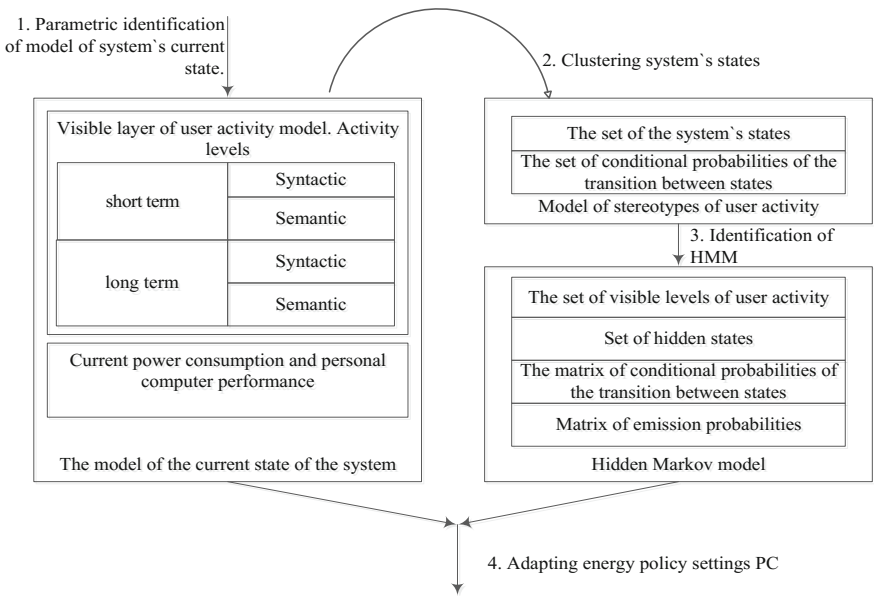


Fig. 2 The interaction of a detailed stochastic model



### 4.1 The Model of the PC's Current State

This model will be considered as a sequence of instant states  $\Xi = \langle S_k, k = 1 \dots K \rangle$ . Each  $S_k$  state is characterized by energy efficiency

$$En_k = \langle P_k, N_k \rangle, \tag{2}$$

where

- $P_k$  is a current performance;
- $N_k$  is a current power consumption.

A user event activity  $UA_k$ :

$$S_k = \langle UA_k, En_k \rangle. \tag{3}$$

User event activity is considered at two levels (syntactic and semantic) and in two intervals: short and long term:

$$UA_k = \langle \underline{Syn}_k, \overline{Syn}_k, \underline{Sem}_k, \overline{Sem}_k \rangle, \tag{4}$$

where

- $\underline{Syn}_k$  is a syntactic level of the user's interaction with the keyboard, mouse, clipboard, etc. on the short-term interval;
- $\overline{Syn}_k$  is a syntactic level of user's activity in the long-term interval;
- $\underline{Sem}_k$  is a semantic level of user's activity that characterizes the results of user control of the start/stop of applications, switching the focus, etc. on the short-term interval;
- $\overline{Sem}_k$  is a semantic level of user's activity in the long-term interval.

Metrics of user activity on syntactic and on semantic levels are the number of events per unit of time. Then the vertex of the stereotypes' model is a cluster, a subset of "close to each other" objects of the A vertex. The distance  $\rho(s_i, s_j)$  between the objects  $S_i$  and  $S_j$  depends on the accepted metric in the space of characteristics.

### 4.2 The Stereotype Model

The stereotype model is a marked graph of state space that characterizes the multiplicity of trajectories in the state of the system's space:

$$MSt = \langle V, E \rangle, \tag{5}$$

where

- $V = \{v_{i,l}\}$  a multiplicity of ordered by the vertices levels (the system's states).
- $E = \{\langle v_{i-1,l}, v_{i,m}, Pr_{i-1,l,i,m} \rangle\}$  the multiplicity of edges that connect the clusters of the previous layer with the clusters of the next layer with the calculated transition probability.

Each vertex contains information about the kernel of the cluster in the model's components of the system's current state:

$$v_{i,l} = \langle \underline{Syn}_{i,l}, \overline{Syn}_{i,l}, \underline{Sem}_{i,l}, \overline{Sem}_{i,l}, P_{i,l}, N_{i,l} \rangle \tag{6}$$

The theory of clustering is used to automatically determine the ratio of objects to the model of the system's current state to the stereotype. The object of data for clustering is the current state ( $S_k$ ). A vector of characteristics is identified with each object  $S_k = \langle \underline{Syn}_k, \overline{Syn}_k, \underline{Sem}_k, \overline{Sem}_k, P_k, N_k \rangle$ ,  $S_k$  components are the separate object characteristics. The number of characteristics is the size of characteristics' space. The multiplicity consisting of all characteristics' vectors we mark as  $A$ :  $A = (s_1, \dots, s_m)$ .

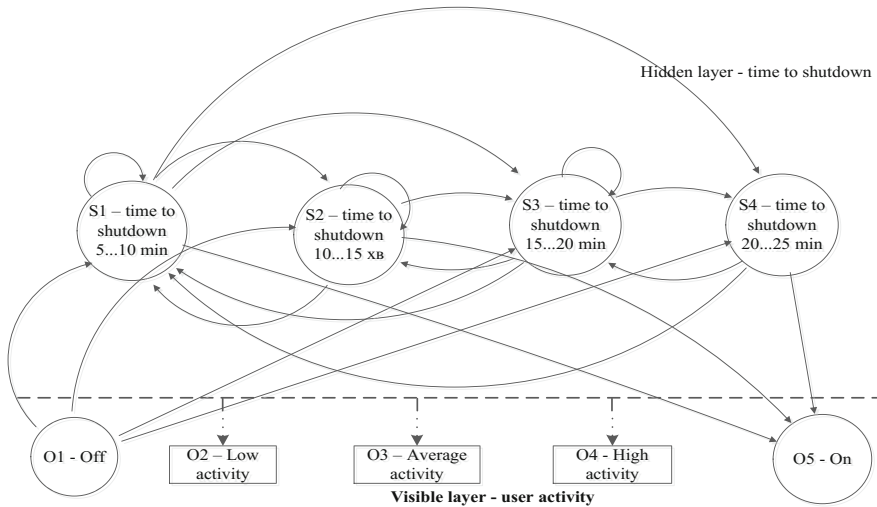
### 4.3 The User Activity Model

The user activity model is based on the hidden Markov model (Fig. 3):

$$MAU = \langle S, O, A, B \rangle, \tag{7}$$

where

- $S$  is a multiplicity of hidden states,  $S = \{S_1, S_2, S_3, S_4\}$ , which characterize the expected time before shutdown (shutdown of a personal computer);
- $O$  is the multiplicity of user activity's visible levels,  $O = \{O_1, O_2, O_3, O_4, O_5\}$  (the visible layer of the hidden Markov model is given by variables, which are measured by the intensity of events for a certain time interval, taking into account the group of data used, also in the visible layer passed two states—the beginning and the end of work, when the user activity is zero);
- $A = \{a_{ij}\} = P(S_t = j | S_{t-1} = i)$  matrix of conditional probabilities of the transition between states  $S$  at the time  $t$ ;
- $B$  is the matrix of emission probabilities.



**Fig. 3** The structure of the hidden Markov model

The event activity of MU user is considered on two levels: syntactic and semantic:

$$MU = \langle Sy, Se \rangle. \tag{8}$$

The method of a personal computer’s adaptive power control system is developed. The key processes in this case are parametric identification of the system’s current state model, clustering of system states, identification of the parameters of the user activity model, adaptation of the PC power consumption policy parameters (Fig. 2).

#### 4.4 Parametric Identification of Models of System’s Current State

The model of the system’s current state involves the identification the parameters of two components: the user’s activity and the current consumption and productivity of the personal computer. Let’s consider the results of observations of a random variable  $S_k$ , that are presented in the form of statistical series. Let’s divide the whole range of available values  $S_k$  at the intervals and count the number of values per each

$i$ -th digit, divide the resulting number by the total number of observations  $n$  and define the frequency that corresponds to a different digit:

$$p_i = \frac{m_i}{n}. \quad (9)$$

To calculate the number of histogram's intervals empirical formula of Sturges is used:  $m \approx (1 + 3, 322 \lg n)$ .

Approximation of histograms, which are obtained experimentally, has an approximate nature and the solution of practical compliance of theoretical histogram's function is based on the criteria for approval of Kolmogorov-Smirnov, Anderson-Darling,  $\chi^2$ .

When solving practical tasks on analysis of real user event activity, the mathematical law of its distribution is unknown.

In this case, an assumption of compliance with the theoretical law of experimental data are extended. This hypothesis requires the statistical test, as a result it either confirmed or refuted.

#### 4.5 Clustering System's States

Given that:  $\Xi$ —a set of vector characteristics,  $\Xi = \langle \langle S_k \rangle, k = 1..K \rangle$ ;  $V$ —set of clusters labels,  $V = \{v_{i,1}\}$ . The vector of characteristics (object)  $S = \langle UA, En \rangle$ —the unit of data for clustering algorithm. It is the element of 6-dimension space:  $S = \langle \underline{Syn}, \overline{Syn}, \underline{Sem}, \overline{Sem}_k, P, N \rangle$ . The characteristic (attribute)  $S_k$  is the component of the scalar vector  $S$ . The dimension 6 is the number of object's characteristics  $S$ .  $k$ -th object with  $S$  is defined as  $S_k = (s_{k,1}, \dots, s_{k,d})$ . The required distance function between objects is  $\rho(s_i, s_j)$ .

We need to build an optimal partitioning of objects into groups i.e. split the sample into subsets that do not intersect which are called clusters, so that each cluster consists of objects that are close in the metric  $\rho$ , and objects from different clusters significantly differ. Thus each object  $s_i \in S$  top model graph of stereotypes is attributed  $v_i \in V$ . The optimality of partitioning is defined as a requirement to minimize mean square error of breakdown (minimizing of the average distance inside the clusters):

$$F_1 = \sum_{v \in V} \frac{1}{|V|} \sum_{s \in V} \rho^2(s, \mu_v) \rightarrow \min, \quad (10)$$

where  $\mu_v$ —is the center of the  $v$  (centroid).

It is necessary to maximize the distance between clusters:

$$F_2 = \sum_{v \in V} \rho^2(\mu_v, \mu) \rightarrow \max, \tag{11}$$

where  $\mu$ —is the common center of mass of the whole sample.

To consider the approximation to the minimum and maximum let's point to a minimum the ratio:

$$\phi = \frac{F_1}{F_2} \rightarrow \min. \tag{12}$$

In this case we take into account limits on the number of clusters because in the optimal case, each object will be a separate cluster.

The solution of the problem. To determine an object belonging to the cluster the method of k-means++ is used.

The first stage: By T steps in the model of the current state of the system  $S = \langle UA, En \rangle$  the sequence of objects is generated  $S_k = \langle \overline{Syn}_k, \overline{Syn}_k, \overline{Sem}_k, \overline{Sem}_i, P_k, N_k \rangle$ , which is the object with characteristics. The input data are a set top model of the current state of the system (S) and the anticipated number of clusters (the number of graph's stereotypes models) N. We will be guided in the normalization not by extreme values, but by the typical ones, statistical characteristics of data such as mean and dispersion:

$$\tilde{s}_i = \frac{s_i - \bar{s}_i}{\sigma_i}; \quad \bar{s}_i = \frac{1}{n} \sum_{a=1}^n s_i^a; \quad \sigma_i^2 = \frac{1}{n-1} \sum_{a=1}^n (s_i^a - \bar{s}_i^a)^2, \quad i = 1, \dots, k. \tag{13}$$

where

- $\tilde{s}_i$  a new item row;
- $s_i$  an element of output row;
- $\bar{s}$  the average row value;
- $\sigma_i$  dispersion;
- $n$  the number of row items.

To limit the range of data the nonlinear transformation is used:

$$\tilde{s}_i = f\left(\frac{s_i - \bar{s}_i}{\sigma_i}\right), \tag{14}$$

where  $f(a) = \frac{1}{1+e^{-a}}$ , which guarantees placement of data in a range [0,1].

The second stage. Since the space is multidimensional, and variable characteristics are normalized, that is increasing impact of dispersions between certain characteristics are not necessary, the Euclidean distance is used:

$$\rho = \sqrt{\sum_{i=1}^D (s_i - \bar{s}_i)^2}, \quad (15)$$

where

$s_i$  is  $i$ -th characteristic of the system  $s$ ;

$\bar{s}_i$  is  $i$ -th characteristic of the system  $\bar{s}$  ( $s \neq \bar{s}$ ),  $i = 1 \dots D$ , where  $D = 6$ —number of characteristics.

The third stage. One of the  $k$ -means problem is the initial cluster centers choice. The solution of this problem is proposed in the modified method  $k$ -means++.

The sequence of the method  $k$ -means++.

Step 1. Select the first cluster center  $c_1$  randomly (among all states of the system  $S$ ).

Step 2. Choose a new cluster center  $c_i$  from  $s \in S$ , for which the square of the distance to the selected centroids will be the largest:

$$\frac{D(s)^2}{\sum_{s \in S} D(s)^2}. \quad (16)$$

Repeat step 2 until you select  $N$  centers of the clusters.

Step 3. For each  $i \in \{1, \dots, K\}$  select the cluster  $C_i$  as a set of points with  $S$ , which are closer to  $c_n$ , than to  $c_j$  for all  $n \neq j$ . The measure of proximity is Euclidean distance between objects (states of the system) and centroids (cluster centers):

$$C_i = \min_n \sqrt{\sum_{p=1}^D (s_{k,p} - c_{n,p})^2}. \quad (17)$$

Step 4. For each  $i \in \{1, \dots, N\}$ , enumerate  $c_i$ , as the center of mass at all points in  $C_i$ :

$$c_i = \frac{1}{|C_i|} \sum_{s \in C_i} s. \quad (18)$$

Step 5. Repeat steps 3–4 until the clusters will change, that is objects will move between clusters or functional change will be minimal  $\phi$ .

### 4.6 Method of Parameter Identification User's Activity Model

The observed value is known (user's activity)  $N(t)$  as a sequence  $N_T = N_0, N_1, \dots, N_{T-1}$  length  $T$ , generated by the sequence of time states to the end of user's experience  $O = O_1, O_2, \dots, O_T$ . Probability to watch  $N(t)$  is  $P(N_T) = \sum_X P(N_T|X_T) P(X_T)$ , where the sum is given by all possible hidden variables  $x(t)$  as a sequence of hidden nodes  $X_T = X_0, X_1, \dots, X_{T-1}$ . According to available data  $N(t)$  it is necessary to determine hidden parameters of the most probable sequence of Markov chain states  $S = S_1 \dots S_T$ .

Thus, if an obtained visible learning sequence of the output power  $O$  is such that  $O = O_1, O_2, \dots, O_T$  ( $O_i \in N_{\text{expr}}$ ), then the parameters of the model  $MAU = (P, B, \pi)$  should maximize a posteriori probability  $p(O|MAU)$ .

The solution of the problem. For definition of the model's parameters  $MAU = (P, B, \pi)$  in this paper iterative procedure which is based on the method of Baum-Welch is used.

First stage: By steps  $T$  in the model  $MAU = (P, B, \pi)$  the sequence of observations is generated  $O = O_1, O_2, \dots, O_T$  ( $O_i \in N_{\text{H3M}}$ ), moreover at time  $t$  for state  $s$ :  $X_t = i$  probability  $P(O_{1,t-1}|X_t = s)$  that during the conversion the sequence of observations was formed  $O_{1,t-1}$ , a  $P(O_{t,T}|X_t = s)$ —the probability that after this condition there is a sequence of observations  $O_{t,T}$ . The probability  $P(X_t = s|O) = P(X_t = s|O_{1,t-1} \cap O_{t,T})$  at time  $t$  the chain is in a state  $s$  is searched.

For any  $s$  state into a random  $t$  step probability  $P(O_{1,t}|X_t = s_i)$  the fact that the way it made to sequence  $O_{1,t}$  for the next  $t$  can be recurrently calculated:

$$\begin{aligned}
 P(O_{1,t}|X_t = s_i) &= \sum_{j \in S} P(O_{1,t}|X_t = s \cap X_{t-1} = j) \\
 &= \sum_{j \in S} P(O_{1,t-1}|X_{t-1} = j)P(X_t = s|X_{t-1} = j)P(O_t = o_t|X_t = s).
 \end{aligned}
 \tag{19}$$

The probability to get into the state  $S$  at the  $t$ -th step, given that after the transition event occurs  $O_t$ , will be equal to the probability to be in a state  $j$  at  $t$ -th step, multiplied by the probability to go out of the state  $j$  at  $s$ , having realized an event  $O_t$  for all  $j \in S$ .

The probability that after a random state  $S$  the sequence will be made  $O_{t+1,T}$  is defined recurrently:

$$P(O_{t,T}|X_t = s) = \sum_{j \in S} P(O_{t+1,T}|X_{t+1} = j)P(X_{t+1} = j|X_t = s)P(o_{t+1}|X_t = s).
 \tag{20}$$

To find the probability that the chain of events will be held,  $P(O)$ , the product should summarize both probabilities for all states at an arbitrary step  $t$ :

$$P(O) = \sum_{s \in S} P(O_{1,t}|X_t = s)P(O_{t,T}|X_t = s), \quad (21)$$

As the future of Markov chain does not depend on the past and the observation of probability events  $O_t$  does not depend on the past observations on the sequence  $O_{1,t-1}$ , then the probability that at time  $t$  the chain will be in a position  $s$ :

$$P(X_t = s|O) = P(X_t = s|O_{1,t-1} \cap O_{t,T}) = \frac{P(X_t = s|O_{1,t-1})P(X_{t+1} = s|O_{t,T})}{P(O)}. \quad (22)$$

The second stage: For a given HMM  $\theta = (P, B, \pi)$  with space of states  $S = \{s_1, s_2, \dots, s_K\}$ , initial probabilities  $\pi_i$  of being in a state  $i$  and probabilities  $P_{i,j}$  of transition from state  $i$  to state  $j$ , observed by  $y_1, \dots, y_T$  and a plurality of output information  $J_{in}$  the most plausible sequence of states of hidden nodes is searched  $S = S_1 \dots S_T$  (Viterbi way) that most accurately describe this model. Then the most likely sequence of states  $x_1, \dots, x_T$  given recurrent relation:

$$V_{1,k} = P(y_1|k)\pi_k, V_{t,k} = P(y_t|k) \max_{x \in S} (P_{x,k} V_{t-1,x}), \quad (23)$$

where  $V_{t,k}$ —the probability of the most plausible sequence of states responsible for the appearance of the first  $t$  visible symbols, culminating in a state  $k$ .

Viterbi path is searched by states  $x$ , satisfying the following equation:

$$x_T = \arg \max_{x \in S} (V_{T,x}), \quad (24)$$

$$x_{t-1} = \begin{cases} \arg(V_{t,k}), & \text{если } t > 1, \\ k, & \text{если } t = 1. \end{cases} \quad (25)$$

The third stage: according to the starting sequence of observations  $O$  the unknown parameters HMM  $\theta$  are determined, that maximize the probability of observing  $O$ :

$$MAU^* = \max_{MAU} (P(O|MAU)). \quad (26)$$

Number of points in time  $r$ , in which observations are carried out is pre-set and consists of the following steps:

- (1) determining the sequence of model states  $S_i = \{s_{i1}, \dots, s_{ir}\}$ ,  $i = 1, \dots, r$  system at a given time;
- (2) assessment of probabilities  $P(S_i)$  of each sequence appearance  $S_i$ ,  $i = 1, \dots, r$ .  
Discovered in the previous step, by calculating the products of probabilities of



transitions between states for the model ranges, limited by control set time points, namely:

$$P(S_i) = \prod_{t=1}^{r-1} P_{s,i,t,t+1}, \tag{27}$$

where  $P_{s,i,t,t+1}$ —the probability of transition from state  $S_{it}$ , in which the system was at time  $t$ , to the state  $S_{i,t+1}$ , at the moment  $t + 1$ ;

- (3) the probability of appearance of the observed sequence  $X = \{x_1, \dots, x_r\}$  for sequences of states  $S_i, i = 1, \dots, r$ :

$$P_{ix} = P(S_i) \prod_{j=1}^r P_{x,i,j}, \tag{28}$$

where  $P_{x,i,j}$ —probability of observable characteristics  $x_j$  in stage  $s_{ij}$ ;

- (4) selection of the most probable sequence of states  $S_{\max} \in \{S_i\}_{i=1, \dots, r}$ , which corresponds the largest probability:

$$\{P\}_{x,\max} = \max_i P_{ix}, \quad i = 1, \dots, r. \tag{29}$$

### 4.7 The Method of PC's Power Consumption Adaptation

It is given: MSt is a model of stereotypes of user's activity, which includes a plurality of states of the system and a set of conditional probabilities of transition between states. The power consumption policy:

$$PM = \langle t, ST \rangle, \tag{30}$$

where

- $t$  is time to finish of PC work;
- ST stage of ACPI, which the system should move to at the end of time  $t$

It is necessary to determine whether the downtime (Tbe) is long enough to offset the overhead of switching between states considering fines for lost productivity. To adapt the parameter  $t$  of the power consumption policies to maximize P performance while minimizing the power consumption of N it is required to minimize the ratio:

$$\sum_{i=1}^n \int_0^{\Delta} \hat{f}_i(\tau) \cdot p(S_t = i) \cdot u(\tau) d\tau \rightarrow \min, \quad (31)$$

where

- n            the number of the system's states;
- $\Delta$          time of transition from active to passive state;
- $u(\tau)$      exponential function of the fine for premature loss of productivity;
- $p(S_t = i)$  the probability of finding a state  $S_t$ ;
- $\hat{f}_i(\tau)$     experimental time that the system spent in one of the states.

Step 1. Define a user session end with a PC via conditional probabilities of model of E stereotypes. For each time point  $t$  gets the probability of being in a state  $V$  for model of stereotypes of user activity. For the system probability of transition from state to state will be defined as:

$$c_t = \max\{P(E_i = t)\}. \quad (32)$$

Step 2. The general approach to reducing the time for the transition to sleep mode is that if the probability of the transition to the standby mode is greater than the given threshold, then the time to standby mode is reduced. Reducing the time of transition to standby mode  $pol(t) = Standby$  occurs when the functional (31) is less than  $\rho$  (admissibility adjustment factor). If the calculated value is greater than  $\rho$ , the time does not change.

Step 3. Check  $t = 0$ , If "yes" there is a transition to a state of sleep, conversely go to step 1.

## 5 Simulation

Physical experiment is carried out in one of the computer classes of the software engineering department of NAU "KhAI". This class has 16 similar types of desktop configuration. Computers' configuration:

- CPU: Intel Core i5-4690;
- GPU: Intel HD Graphics 4600;
- HDD: 1 Tb;
- OS: Microsoft Windows 7 Service Pack 1 64-bit;
- RAM: 8 Gb.

**Data collection technologies.** Hooking is the ability to hook in the OS's messaging system, which allows you to track a particular message (including an event) until it reaches the target function.

As a mechanism for hooking, we use the functionality of the operating system of the Windows family (hereafter OS Windows) «WindowsHook».

Thus, the method of hooking, based on an event model, is implemented next way.

- Step 1. To register the event handlers in the event manager. In response to the external influence, generate an event.
- Step 2. In response to an external event, a Windows system event is generated.
- Step 3. To send the event to hooker.
- Step 4. The hooker takes an event and looks for a handler for it, written beforehand.
- Step 5. The event handler processes the event, saves the event data.
- Step 6. The event is further transmitted in the event manager.
- Step 7. The event manager takes an event and looks for a handler for it.
- Step 8. The event manager calls the handler.
- Step 9. If the method work isn't terminated then to go to step 2, else to finish.

Methods for obtaining data on ACPI states, transition times between states and power consumption in the current state, the appropriate tool for obtaining the required parameters are Intel Power Gadget.

A software prototype to collect and store user activity information was developed on the basis of the Hookings mechanism.

The prototype of user activity data collection software is intended for:

- tracking and recording input/output events;
- tracking and registration the start and stop of applications;
- definition of running and stopped applications at a given time point;
- saving data in xml format in the form of a certain structure.

Since the experiment was conducted in the computer classroom, the following characteristic of sequences have been singled out based on theoretical assumptions that the user's PC activity in the classroom depends on the learning process, as well as on the basis of experimental data:

- the lesson in the middle of the semester (Class A). This sequence can be interpreted as: low activity at the beginning of the lesson due to the fact that the teacher conducts introductory acquaintance in order to work at the lesson, high activity during work at the lesson changes with average activity, as well as low activity during the break between the half-periods and at the end of the lesson. An example of a sequence: {01-02-02-03-05-05-05-02-01-01-05-05-06-01-01-04-04-02-01};

- the lesson before the session (Class B). Example: {04-05-05-06-05-05-03-01-04-04-05-04-04-05-06-06-05-05-01}, such a sequence of activity is specific for a lesson before the session, the defense of a diploma or projects. You can notice that the user activity is high throughout the lesson and there are only a few changes during short breaks and consultations with the teacher;
- the lesson at the beginning of the semester (Class C), an example of a PC user activity sequence: {02-01-01-01-03-04-05-06-04-01-01-05-04-04-04-04-02-01}. Such class is typical when it takes more time to work with the theoretical material at the beginning of the lesson, and during the lesson there is a reduced activity of the user, connected with the state of students after the pause in the educational process;
- the consultation lesson (Class D) {01-01-01-01-05-05-05-01-01-01-06-06-05-06-01-01-02-01-03}, during which the PC is used from time to time, that is, there are no clearly selected transitions between the lowest and the very high activity.

For the experimental evaluation of the trained user activity model based on the collected statistical data during the experiment, we will classify the user activity sequences on the test data sample. As indicators of the adequacy of the trained user activity model, the matching factor between the output data and the obtained results was selected, namely the Cohen's kappa coefficient, the Kendall rank correlation coefficient, Pearson correlation,  $\chi^2$  and the overall consistency of the results of the training and the test sample. The Cohen's kappa coefficient is considered more reliable than the simple calculation of the compliance's percentage, because it takes into account the agreement of random variables which occur [24]. Cohen's kappa coefficient measures the agreement between two evaluators, each of which classifies  $N$  elements in  $C$  mutually exclusive categories. Cohen's kappa coefficient  $k$ :

$$k = \frac{p_o - p_e}{1 - p_e} = 1 - \frac{1 - p_o}{1 - p_e}, \quad (33)$$

where

- $p_o$  is the relative observed consistency between evaluators;
- $p_e$  is the hypothetical probability of random coherence, using observable data to calculate the probability of each observer randomly for each category.

If the coefficient  $k = 1$ , then the data are completely consistent. Otherwise, we can talk about consistency only at a certain level of trust.

The Kendall rank correlation coefficient. It is used to identify the relationship between quantitative or qualitative indicators if they can be ranked. The values of the index  $X$  are displayed in ascending order and assigned ranks to them. Ranking the value of  $Y$  and count the Kendall correlation coefficient [25]:

$$\tau = \frac{2S}{n(n - 1)}, \tag{34}$$

where  $S = P-Q$ ;

- P is the total number of observations followed after the current observations with a large rank values Y;
- Q is the total number of observations followed by current observations with lower rank values Y;
- n is the number of elements in a row.

The Pearson correlation coefficient characterizes the existence of a linear relationship between two values [26]. The Pearson correlation coefficient is calculated by the formula (35):

$$r_{xy} = \frac{\sum_{i=1}^m (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^m (x_i - \bar{x})^2 \sum_{i=1}^m (y_i - \bar{y})^2}} = \frac{\text{cov}(x, y)}{\sqrt{s_x^2 s_y^2}}, \tag{35}$$

where

- $\bar{x}, \bar{y}$  are the selective averages  $x^m$  and  $y^m$ ;
- $s_x^2, s_y^2$  are the selective dispersions.

Mathematical modeling of training algorithms and classification is performed using the open source software [27].

The calculation of model parameters is carried out using the method of parametric identification of the user’s activity model. The purpose of the calculation is to determine the matrix of transition probabilities’ distribution  $\mathbf{P} = \{p_{ij}\}$ , matrix  $\mathbf{B}$  distribution of probabilities of user activity’s observed levels.

**Table 1** Output data for learning the user activity model

Sequence	Label	Hidden states
0-1-1-2-4-4-4-1-0-0-4-4-5-0-0-3-3-2-0	A	4
0-0-1-2-4-4-4-1-0-0-4-4-5-0-0-3-3-1-0	A	4
0-1-0-3-4-4-4-1-0-0-4-4-5-0-0-3-3-0-0	A	4
0-0-0-4-4-4-4-1-0-1-4-4-5-0-0-3-4-0-0	A	4
3-4-4-5-4-4-2-0-3-3-4-3-3-4-5-5-4-4-0	B	4
4-4-4-4-4-4-4-2-2-0-4-4-5-3-3-4-5-4-5	B	4
3-4-4-4-4-5-5-1-0-2-4-4-5-0-0-4-4-4-4	B	4
1-0-0-0-2-3-4-5-3-0-0-4-3-3-3-3-1-0	C	4
0-0-1-1-2-2-2-3-0-0-2-3-4-2-1-2-2-1-1	C	4
0-0-0-0-0-0-1-2-3-3-2-1-0-1-2-4-0-0-0	C	4
0-0-0-0-0-4-4-4-5-1-1-0-0-5-5-4-5-0-0	D	4
0-0-4-4-4-0-0-0-5-5-4-4-0-1-0-4-5-0	D	4
4-4-5-0-0-0-3-5-4-4-0-1-0-0-4-4-4-4-5	D	4

To study the user activity model, the most common sequences of user activity levels were used for the lesson in NAU “KhAI”. They were observed for the different classes of activity which were given earlier. Output data are shown in Table 1.

## 6 Results and Analysis

The adequacy of the model is verified using parametric and non-parametric statistics. We can note that the concept of adequacy in this case was interpreted as a criterion for the accuracy of the activity class definition with the help of a trained model based on test sets of data. Test results are shown in Fig. 4.

Evaluating the adequacy of parametric statistics, we find that the difference between the initial data and their relation to the corresponding class does not exceed 1%.

In order to detect the differences in power consumption while PC working in the classroom, a prototype of the PC’s adaptive power control was used. With it the modeling on the basis of experimental data collected from the PC was carried out. We also give a comparison of the average energy consumed on a similar machine of the same computer class during the working day.

Figure 5 shows the PCs consumption with standard policy and adaptive control policy.

The main advantage in power consumption is taken for 4 h of PC operation, The main gain for energy consumption is at 4 h of the PC work, it can be correlated with

Sequence	True Class	Assigned Class	Likelihood	Match
0-1-1-2-4-4-4-1-0-0-4-4-5...	A	A	0.995412540842886	<input checked="" type="checkbox"/>
0-0-1-2-4-4-4-1-0-0-4-4-5...	A	A	0.99889823587246	<input checked="" type="checkbox"/>
0-1-0-3-4-4-4-1-0-0-4-4-5...	A	A	0.996136963002419	<input checked="" type="checkbox"/>
0-0-0-4-4-4-4-1-0-1-4-4-5...	A	A	0.958548650279292	<input checked="" type="checkbox"/>
3-4-4-5-4-4-2-0-3-3-4-3-3...	B	B	0.999278249668276	<input checked="" type="checkbox"/>
4-4-4-4-4-4-2-2-0-4-4-5...	B	B	1	<input checked="" type="checkbox"/>
3-4-4-4-4-5-5-1-0-2-4-4-5...	B	B	0.922877645427977	<input checked="" type="checkbox"/>
1-0-0-2-3-4-5-3-0-0-4-3...	C	C	0.966992631865798	<input checked="" type="checkbox"/>
0-0-1-1-2-2-3-0-0-2-3-4...	C	C	0.999998404120797	<input checked="" type="checkbox"/>
0-0-0-0-0-1-2-3-3-2-1-0...	C	C	0.998202684030266	<input checked="" type="checkbox"/>
0-0-0-0-4-4-4-5-1-1-0-0...	D	D	0.986329395996206	<input checked="" type="checkbox"/>
0-0-4-4-4-0-0-0-0-5-5-4-4...	D	D	0.979550726843901	<input checked="" type="checkbox"/>
4-4-5-0-0-3-5-4-4-0-1-0...	D	D	0.992585024284452	<input checked="" type="checkbox"/>
**				<input type="checkbox"/>

Проверить

Показатели эффективности																		
Number of samples	Number of classes	Maximum Hits	Minimum Hits	Kappa Coefficient (k)	Kappa (k) Std. Error	Kappa (k) Variance	Kappa (k) H. Variance	Kappa (k) H. Std. Error	Tau Coefficient (τ)	Pearson Chi-square (χ²)	chi-Square (χ²)	chiuprow T	Pearson's C	Sakoda's V	Cramér's V	Overall igreemen	geometric greemen	Chance greemen
13	4	4	3	1	5,5432...	3,0727...	0,0259...	0,1609...	1	1,7320...	39	0,2107...	0,8660...	1	1	1	3,2237...	0,2544...

Fig. 4 The window of the trained model test results

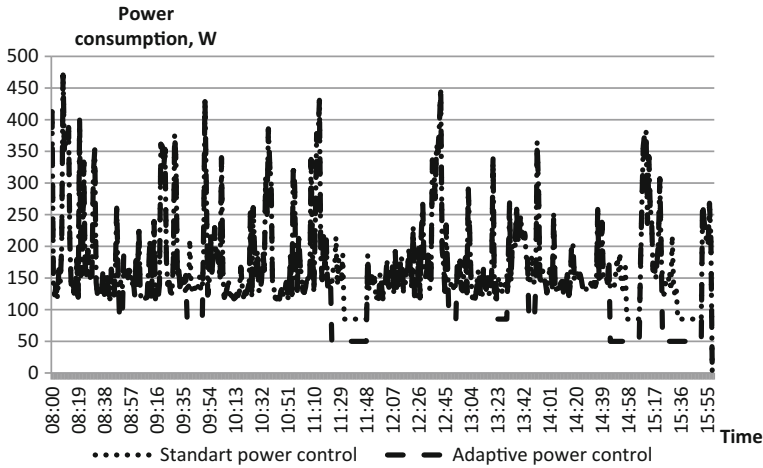
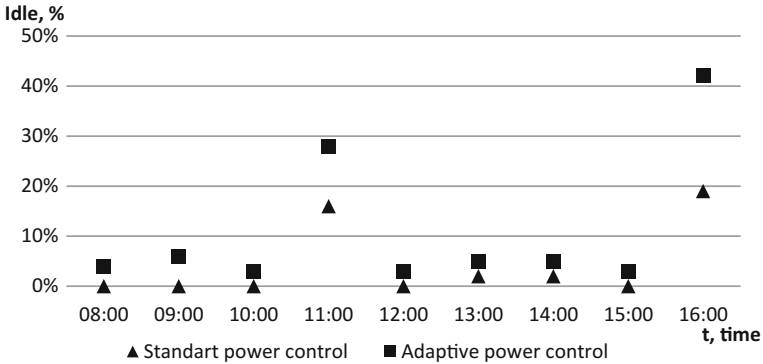


Fig. 5 Power consumption by a personal computer with different power control technologies

the fact that in the interval from 11.25 to 11.55 there is a break in the students schedule, under standard settings, the PC switched to the level S1 sleep stage, after 10 min of it’s inactivity. With adaptive power consumption policies, the PC went into sleep after 3 min of inactivity, with at lower power consumption ACPI S2. At the 8th hour of working, when the students’ lessons end at 15.25, and working day at 16.00, PC switched to S1 sleep state after 10 min of inactivity on basic settings of the time out policy. With adaptive policy it switched to S3 sleep state after 3 min, since user activity at the end of the 4th lesson was low, which reduced the penalty for loss of productivity and increased the probability of transition to a state of reduced power consumption for a longer period. It allowed to reduce power consumption for 1 PC by 0.7 kWh. Taking into account that in the computer class there are 18 computers, the savings can be about 12.6 kWh.

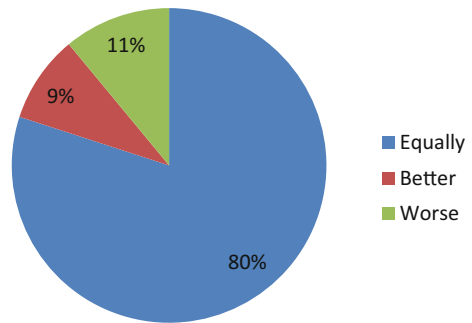
A comparison of hourly percent of idle time between two power control technologies is shown in the Fig. 6. The peculiarity of the user’s work for the PC in the period of time in the middle of the study semester are frequent breaks, but they are less than time policy of time out. For adaptive politics this period is characterized by the average and low user’s activity, which is characteristic for the transferring of the PC in less time intervals to sleep state, with the sleep state of the first S1 level. This state allows going into working condition for 2–5 s, while consuming 4 times less electricity. Sleep state time for adaptive power control was 1 h per day, and for basic technology—23 min.

In order to evaluate the differences in the comfort of working with a personal computer using the basic technology of power control and adaptive power control of the PC, a ranked anonymous interview was conducted. Students were in the role



**Fig. 6** The percentage of PC's idle time per hour with different power consumption policies in the middle of the semester

**Fig. 7** Subjective valuation of the PC's comfortable work with adaptive power control, unlike basic technology



of experts who were asked to evaluate the comfort of working with the PC after the end of the working day. The questionnaire contained one question: “Evaluate the comfort of using the PC today compared with yesterday?”. Possible answers: “Better (2), Equally (1), Worse (0)”. To determine the coordinated degree of 200 experts’ opinion, a special measure is used—Kendall concordation coefficient ( $W$ ). As a result,  $W = 0,02$ , it follows that the differences between the opinions of experts are insignificant. The result of the interview is presented in Fig. 7.

Thus the electricity savings for the semester for one computer class with 16 (computer’s amounted to 1025 kWh, which in monetary terms (in June 2016) was 25 UAH (~ 1 \$) per every computer of the educational class every month, while the permissible indicators of the PC’s using were saved.

In valuating the effectiveness of PC’s adaptive power control, the metric of cash resources’ savings by reducing the cost of replacing the components of the PC was not taken into account, as the amount of time spent on the worked hours for failure decreases. It is worth noting that, in addition to the material efficiency indicator, a reduction in the negative impact of the PC on the environment is achieved.



## 7 Conclusion

The main results include a method of adaptive power management of the personal computer, which allows you to achieve lower energy costs of the personal computer while maintaining acceptable performance comfort to the user. PC usage implemented a sequence of four steps.

1. Parametric identification of models in the exact state of the system is performed by methods of mathematical statistics and scores from the previous processing.
2. Clustering states of the system are based on the method k-means++ and allow to reduce the dimension and complexity of the problem.
3. Identification of hidden Markov model of user activity uses the Baum-Welch method and can be attributed to the current user behavior to one of the elements of the set of stereotypes user's activities.
4. The adaptation of the policy settings PC energy forms the management solutions considering of fines for loss of productivity and overheads.

## References

1. Mingay, S.: Green IT: a new industry shock wave. Gartner, pp. 1–28. [online] Available at: [http://www.ictliteracy.info/rtf.pdf/Gartner\\_on\\_Green\\_IT.pdf](http://www.ictliteracy.info/rtf.pdf/Gartner_on_Green_IT.pdf) (2007). Accessed 14 Oct 2017
2. Uchchukwu, A., Li, K., Shen, Y.: Improving cloud computing energy efficiency. In: 2012 IEEE Asia Pacific Cloud Computing Congress (APCloudCC) (2012)
3. Download.intel.com.: Enhanced Intel® SpeedStep® Technology for the Intel® Pentium® M Processor. [online] Available at: <http://download.intel.com/design/network/papers/30117401.pdf> (2017). Accessed 7 Oct 2017
4. Phoronix.com. Intel EIST SpeedStep—Phoronix. [online] Available at: <https://www.phoronix.com/scan.php?page=article&item=397&num=1> (2017). Accessed 12 Oct 2017
5. KV.by. Cool'n'Quiet. [online] Available at: <https://www.kv.by/archive/index2005030502.htm> (2017). Accessed 13 Nov. 2017
6. Snowdon, D., Ruocco, S., Heiser, G.: Power management and dynamic voltage scaling: myths and facts. *Nat ICT Aust. Sch. Comput. Sci. Eng.* **12**, 1–7 (2005)
7. . Acpi.info.: ACPI—Advanced Configuration and Power Interface. [online] Available at: <http://www.acpi.info> (2017). Accessed 8 Nov 2017
8. Benini, L., Micheli, G.: *Dynamic power management*. Springer, US, Boston, MA (1998)
9. Karlin, A., Manasse, M., McGeoch, L., Owicki, S.: Competitive randomized algorithms for nonuniform problems. *Algorithmica* **11**(6), 542–571 (1994)
10. Helmbold, D., Long, D., Sconyers, T., Sherrod, B.: Adaptive disk spin-down for mobile computers. *Mob. Netw. Appl.* **5**(4), 285–297 (2000)
11. Hwang, C., Wu, A.: A predictive system shutdown method for energy saving of event-driven computation. *ACM Trans. Des. Autom. Electron. Syst.* **5**(2), 226–241 (2000)
12. Chung, E., Benini, L., De Micheli, G.: Dynamic power management using adaptive learning tree. In: *IEEE/ACM International Conference on Computer-Aided Design*, pp. 274–279. IEEE Press (1999)
13. Lu, Y., De Micheli, G.: Adaptive hard disk power management on personal computers. In: *Ninth Great Lakes Symposium on*, pp. 50–53. VLSI (1999)

14. Benini, L., Bogliolo, A., Paleologo, G., De Micheli, G.: Policy optimization for dynamic power management. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **18**, 813–833 (1999). <https://doi.org/10.1109/43.766730>
15. Qiu, Q., Pedram, M.: Dynamic power management based on continuous-time Markov decision processes. In: *Proceedings of the 36th ACM/IEEE Conference on Design Automation Conference—DAC'99* (1999). <https://doi.org/10.1145/309847.309997>
16. Lu, Y., Chung, E., Šimunić, T. et al.: Quantitative comparison of power management algorithms. In: *Proceedings of the Conference on Design, Automation and Test in Europe—DATE'00* (2000). <https://doi.org/10.1145/343647.343688>
17. Simunic, T., de Micheli, G., Benini, L.: Event-driven power management of portable systems. *12th International Symposium on System Synthesis*, pp. 18–23. IEEE Computer Society, Washington (1999)
18. Turkin, I., Vdovitchenko, A.: Energy-efficient scheduling for portable computers as bi-criteria optimization problem. In: Kharchenko, V., Kondratenko, Y., Kacprzyk, J. (eds.) *Green IT Engineering: Concepts, Models, Complex Systems Architectures. Studies in Systems, Decision and Control*, vol 74, pp. 87–100. Springer, Cham (2017). [https://doi.org/10.1007/978-3-319-44162-7\\_5](https://doi.org/10.1007/978-3-319-44162-7_5)
19. Saravjit, R.: Engagement experiences on the path to richer insight and truth. In: *Russian UX Conference*, p. 76 (2014)
20. Kurzantseva, L.: On the construction of intelligent adaptive interface based on agent-based technology for a general-purpose computer systems. *Inf. Technol. Comput. Eng.* **1**, 16–20 (2010)
21. Yu, C., Mannor, S., Theodorou, G., Pfeffer, A.: User model and utility based power management. *AAAI'07: Materials of the 22nd National Conference 1918–1919* (2007)
22. Graybill, R., Melhem, R.: *Power Aware Computing*, pp. 101–125 (2002)
23. Kharchenko, V., Illiashenko, O.: Concepts of green IT engineering: taxonomy, principles and implementation. In: Kharchenko, V., Kondratenko, Y., Kacprzyk, J. (eds.) *Green IT Engineering: Concepts, Models, Complex Systems Architectures. Studies in Systems, Decision and Control*, vol 74, pp. 3–19. Springer, Cham (2017). [https://doi.org/10.1007/978-3-319-44162-7\\_1](https://doi.org/10.1007/978-3-319-44162-7_1)
24. Bakeman, R., Gottman, J.: *Observing Interaction*, pp. 10–25 (2010)
25. Schervish, M.: *Theory of statistics*. Springer, New York, New York, NY (1995)
26. Graham, A.: *Statistics*. Hodder Education, London (2011)
27. Fitzgerald, B.: Open source software adoption. *Int. J. Open Sour. Softw. Process.* **1**, 1–23 (2009). <https://doi.org/10.4018/jossp.2009010101>