

УДК 004

DOI: 10.26102/2310-6018/2019.26.3.016

С.Н. Широбокова, А.А. Кацупеев,
Д.М. Жевакин, О.Н. Сериков, С.П. Воробьев
**О ПОДХОДЕ К РЕАЛИЗАЦИИ ИНТЕЛЛЕКТУАЛЬНОГО
ИНСТРУМЕНТАРИЯ АВТОМАТИЗИРОВАННОГО УЧЕТА
И АНАЛИЗА РАБОЧЕГО ВРЕМЕНИ СОТРУДНИКОВ**

*Южно-Российский государственный политехнический университет
(НПИ) имени М.И. Платова,
Новочеркасск, Россия*

Задача корректного учета рабочего времени сотрудника тесно связана с вопросами контроля трудовой дисциплины и использования рабочего времени, вопросами оценки сроков выполнения работ, учета времени, потраченного на выполнение той или иной задачи (что особенно важно при почасовой оплате, например, при аутсорсинге услуг). Авторами предложен новый подход к автоматизированному анализу интервалов активности и простоя работы сотрудника за компьютером, который заключается в том, что пользователю необходимо время на обдумывание решения задачи, выполнение каких-то подготовительных работ не на компьютере. В статье представлена формализованная постановка задачи, позволяющая учитывать периоды активности и потенциального бездействия при подготовке в полезном рабочем времени. Описаны аспекты практической реализации программного инструментария, основанного на этом подходе. Использовался стек трёх технологий: AutoIt, Python, платформа «1С:Предприятие». С помощью скриптового языка AutoIt реализован перехват действий пользователя и запись в лог-файлы необходимой информации в зашифрованном виде. На языке Python осуществлялся анализ данных, отслеживались пики активности пользователя. На платформе «1С:Предприятие» был реализован пользовательский интерфейс и велось общее управление программой.

Ключевые слова: учет рабочего времени сотрудника, формализованная модель задачи, автоматизированный контроль использования рабочего времени, AutoIt, Python, платформа «1С:Предприятие».

Введение

Одной из актуальных проблем управления предприятием в современных *ERP*-системах является задача корректного учета рабочего времени сотрудника, которая тесно связана не только с вопросами контроля трудовой дисциплины и использования рабочего времени, но и с вопросами оценки сроков выполнения работ, учета времени, потраченного на выполнение той или иной задачи (что особенно важно при почасовой оплате, например, при аутсорсинге услуг). От эффективности решения этой задачи зависит и цена услуги для клиента, и в целом производительность и доходность бизнеса.

Существуют ручной и автоматизированный подходы к решению задачи учета рабочего времени сотрудника, каждый из них имеет свои преимущества и недостатки. В случае использования ручного подхода

сотрудник лично фиксирует момент начала и окончания работ по задаче. Автоматизированный подход предполагает использование специализированной программы, которая отслеживает действия пользователя, проводит их анализ и формирует отчет о том, какие задачи выполнялись и сколько времени это заняло [1-5].

Авторами предложена новая технология анализа интервалов активности и простоя работы сотрудника за компьютером, приведено формализованное описание задачи и рассмотрены аспекты практической реализации программного инструментария, основанного на этой технологии.

Формализованное описание и концепция подхода к решению задачи

Рассматриваемый подход к решению задачи автоматизированного учета и анализа рабочего времени сотрудников базируется на двух аспектах: первый – анализ активности работы за компьютером, второй – это анализ осмысленности и ценности набранного текста. В существующих программных решениях, как правило, анализируется активность пользователя путем перехвата символов с клавиатуры и анализа периодов простоя. Если человек не нажимает клавиши клавиатуры в течение периода, превышающего заданное пороговое значение (например, дольше 5 минут), то этот период времени считается простым. Такой подход к анализу деятельности эффективен для оценки работы обычного оператора, который постоянно занимается однотипными задачами, не требующими осмысления и творческого отношения. Авторами выдвигается предположение, что пользователю необходимо дать время на обдумывание решения задачи, выполнение расчетов на бумаге, вследствие чего активность взаимодействия с компьютером будет низкой, но это не означает, что он не работает над поставленной задачей. Поэтому, если после такого «простоя» за компьютером идет большая активность (написание текста, разработка рисунков и т.д.), то можно считать, что сотрудник все это время работал над задачей.

Суть второго концептуального аспекта решения заключается в том, что необходимо понять, работал ли человек или просто, например, набирал текст некоторой книги (хотя, если это его работа, то почему бы и нет). Поэтому производился перехват слов, которые он набрал с клавиатуры (при этом они шифровались, чтобы не нарушать личное пространство пользователя), и проводился анализ со словарем полезных слов. Данный словарь включал именно ключевые слова, которые характерны для определенной программы и для определенного рода деятельности. Таким образом, логика работы инструментария строится на основе анализа пиков активности и «задумчивости», а также и анализа наличия «полезных» слов.

Для дальнейшей алгоритмизации задачи учета полезного времени работы пользователя за компьютером использовалась модель, описанная ниже.

Есть множество проектов, над которыми ведут работу сотрудники организации:

$$Projects = |Project^q|, q = \overline{1, Q}.$$

Элемент множества, описывающий q -й проект, формально можно представить следующим образом:

$$Project^q = \langle NameProject^q, TasksProject^q \rangle,$$

где $NameProject^q$ – наименование q -го проекта, $TasksProject^q$ – массив задач q -го проекта.

Массив задач проекта описывается следующим образом:

$$TasksProject^q = |TaskProject_p^q|, p = \overline{1, P^q}.$$

Элемент множества, описывающий задачу проекта, формально можно представить следующим образом:

$$TaskProject_p^q = \langle NameTask^{pq}, KeyWordsTask^{pq}, Program^{pq} \rangle,$$

где $NameTask^{pq}$ – наименование p -ой задачи q -го проекта; $KeyWordsTask^{pq}$ – множество слов, идентифицирующих выполнение p -ой задачи q -го проекта; $Program^{pq}$ – программа, в окне которой пользователь выполняет p -ую задачу q -го проекта.

Множество программ, используемых для выполнения проектов, можно представить как:

$$Programs = |Program^r|, r = \overline{1, R}.$$

Каждый элемент множества, описывающий r -ую программу, формально можно описать следующим образом:

$$Program^r = \langle NameProgram^r, KeyWordsProgram^r \rangle,$$

где $NameProgram^r$ – наименование r -го программного продукта, $KeyWordProgram^r$ – множество слов, употребление которых при работе пользователя в окне программы, свидетельствует о выполнении работы, соответствующей назначению r -ого программного продукта (словарь ключевых «полезных» слов).

Множество «полезных» слов программного продукта описывается следующим образом:

$$KeyWordsProgram^r = |KeyWordProgram_h^r|, h = \overline{1, H^r}. \quad (1)$$

Указанное выше множество $KeyWordsTask^{pq}$ слов,

идентифицирующих выполнение -ой задачи q -го проекта в окне программы $Program^{pq} = Program^r \in Programs$, является подмножеством множества «полезных» слов этого программного продукта:

$$KeyWordsTask^{pq} \subset KeyWordProgram^{pq}$$

Структура информации о действиях -го пользователя из записанного лог-файла может быть представлена как некоторое множество:

$$UsersInformation^s = |UsersInformation^{st}|, t = \overline{1, T^s},$$

в котором каждый элемент представляет собой структуру вида:

$$UsersInformation^{st} = \langle Words^{st}, time^{st} \rangle,$$

где $Words^{st} = |Word_i^{st}|, i = \overline{1, I^t}$ – массив слов, которые были введены s -м пользователем и зарегистрированы на момент времени $time^{st}$.

Структура информации о действиях -го пользователя в окнах определенных программ из записанного лог-файла может быть представлена как некоторое множество:

$$UsersTimeInformation^s = |UsersTimesInformation^{sj}|, j = \overline{1, J^s}, \quad (2)$$

в котором каждый элемент представляет собой структуру вида:

$$UsersTimesInformation^{sj} = \langle Program^{sj}, TimeBegin^{sj}, TimeEnd^{sj}, DurationUsersTimesInformation^{sj} \rangle,$$

где $Program^{sj}$ – j -ая программа, которую использовал -ый пользователь (J^s – количество окон программ, использованных пользователем за анализируемый период рабочего времени),

$TimeBegin^{sj}$ – время начала работы -го пользователя в j -ой программе,

$TimeEnd^{sj}$ – время окончания работы -го пользователя в j -ой программе,

$DurationUsersTimesInformation^{sj}$ – продолжительность периода времени нахождения s -го пользователя в окне определенной программы j .

На этапе анализа двух логов (введенных слов и периодов работы в программных продуктах) проводится сопоставление, в окнах каких программных продуктов был произведен ввод соответствующих слов. Формируется множество $WordInProgram^{sj}$, которое является подмножеством множества $UsersInformation^s$ и включает те элементы, для которых выполняется следующее условие:

$$TimeBegin^{sj} \leq time^{st} < TimeEnd^{sj},$$

т.е. элементы, описывающие набор слов, ввод которых осуществлялся в период работы -го пользователя в j -ом окне программы:

$$WordInProgram^{sj} = |WordInProgram_g^{sj}|, g = \overline{1, G^{sj}}, \quad (3)$$

где G^{sj} – количество слов, введенных s -м пользователем в j -ом окне программы.

Формируется множество $GoodWord_m^{sj}$, которое является подмножеством $WordInProgram^{sj}$ и включает те элементы, для которых выполняется следующее условие:

$$WordInProgram_g^{sj} \in KeyWordProgram^j,$$

т.е. элемент, описывающий набор слов, введенный в j -ом окне программы, входит в состав словаря ключевых «полезных» слов j -ой программы.

$$GoodWords^{sj} = |GoodWord_m^{sj}|, m = \overline{1, M^{sj}},$$

где M^{sj} – количество ключевых «полезных» слов, введенных s -м пользователем в j -ом окне программы.

Структура информации о полезности действиях s -го пользователя в j -ом окне программы, записываемая в итоговый лог-файл, может быть представлена как некоторое множество:

$$UserWorkInformation^{sj} = \langle Program^{sj}, TimeWork^{sj}, G^{sj}, M^{sj} \rangle, (4)$$

где $Program^{sj}$ – j -ое окно программы, в котором работал s -ый пользователь, $TimeWork^{sj}$ – множество, описывающее время работы программы, G^{sj} – общее количество введенных слов в окне программы, M^{sj} – количество полезных слов в окне программы.

Множество описывается следующим образом:

$$TimeWork^{sj} = |TimeWork_k^{sj}|, k = \overline{1, K^{sj}}.$$

Каждый элемент этого множества представляет собой структуру вида:

$$TimeWork_k^{sj} = \langle TTimeWork_k^{sj}, PrTimeWork_k^{sj}, DurationTimeWork_k^{sj} \rangle,$$

где $TTimeWork_k^{sj}$ – отдельный k -й интервал времени работы программы, $PrTimeWork_k^{sj}$ – признак вида активности, $DurationTimeWork_k^{sj}$ – длительность k -го интервала времени активности.

Используется среднее количество слов для определения степени активности пользователя в конкретный период времени посредством сравнения отношения количества полезных слов к среднему количеству полезных слов, набранным пользователем за весь период времени.

На заключительном этапе проводится анализ активности пользователя и распределение по шкале интервалов активности в соответствии со следующим алгоритмом.

Выполняется расчет среднего количества полезных слов, которое вводит пользователь за выбранный период времени $TimeWork_k^{sj}$.

$$AverageWord_k^{sj} = \frac{M_k^{sj}}{TimeWork_k^{sj}}$$

$$AverageWordTotal^{sj} = \frac{M^{sj}}{TimeWork^{sj}}$$

$$UserKA_k = \frac{AverageWord_k^{sj}}{AverageWordTotal^{sj}}$$

Далее проводится анализ отношения количества полезных слов $AverageWord^{sj}$ к среднему количеству полезных слов, вводимых пользователем в выбранный период активности G^{sj} .

$$PrTimeWork_k^{sj} = \begin{cases} UserTimeActivity, & \text{если } UserKA_k \geq KAP \\ UserTimeInactivity, & \text{если } UserKA_k \leq KAP \\ UserTimeThinking, & \text{если } UserKA_k \leq KAP \text{ and } UserKA_{k+1} \geq KAA \end{cases} \quad (5)$$

Результат подобного отношения позволяет отнести данный период рабочего времени $TimeWork_k^{sj}$ к одному из трех видов активности (5): активен $UserTimeActivity$, не активен $UserTimeInactive$, период обдумывания $UserTimeThinking$. Если, например, отношение меньше либо равно $KAP=0.6$ от среднего, а также, если последующий период активности не относится к пику активности, данному периоду активности присваивается статус «не активен», если отношение находится в промежутке от $KAP=0.6$ до $KAA=1.4$ – присваивается статус «активен» и продолжается анализ. В случае отношения большего либо равного $KAA=1.4$ присваивается значение пиковой активности и рамках предыдущей активности пользователя считается, что он обдумывал задачу, и включается в период активного времени работы (Рисунок 1). Дополнительным условием включением интервала в период активной работы является условие, что длительность периода обдумывания не должно составлять, например, более 70% от последующего периода активности.

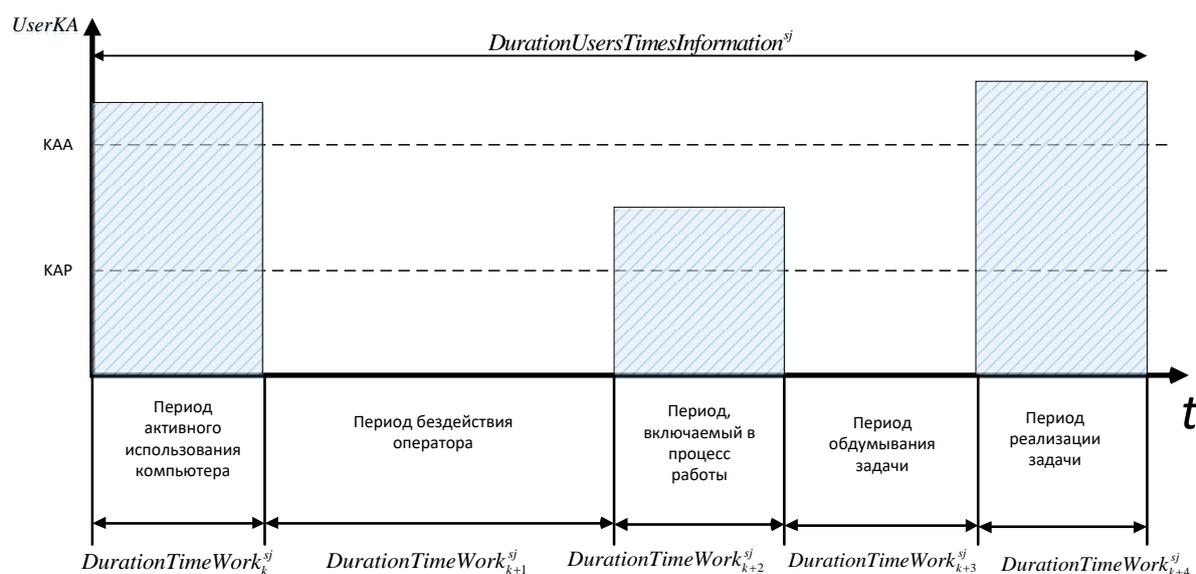


Рисунок 1 – Временная диаграмма работы пользователя

Аспекты программной реализации

Для решения задачи использовался стек из трёх технологий: *AutoIt*, *Python*, платформа «1С:Предприятие» (Рисунок 2):

– *AutoIt* позволяет создавать скрипты автоматизации, способные имитировать действия пользователя, такие как текстовый ввод и воздействия на элементы управления системы и программ.

С помощью скриптового языка *AutoIt* реализован перехват действий пользователя и запись в лог-файлы необходимой информации в зашифрованном виде.

– *Python* – имеет множество библиотек, позволяющих осуществлять проверку на ошибки, а также ряд других действий, связанных с анализом и управлением файлами. На языке *Python* осуществлялся анализ данных, отслеживались пики активности пользователя.

– Среда *1С* используется как управляющая система, в которой возможно запускать скрипты из *AutoIt* и *Python* в фоновом режиме, что позволяет пользователю параллельно выполнять основную работу. На платформе «1С:Предприятие» был реализован пользовательский интерфейс и велось общее управление программой.

На первом этапе считываются данные с нажатий клавиш пользователя, а также отслеживается время нахождения в определенном окне программы. Все это происходит с использованием скрипта, написанного в среде *AutoIt*. Используемая библиотека для считывания данных с клавиатуры *Misc.au3*. Также все данные, которые вводит пользователь, шифруются, чтобы избежать утечки личной информации.

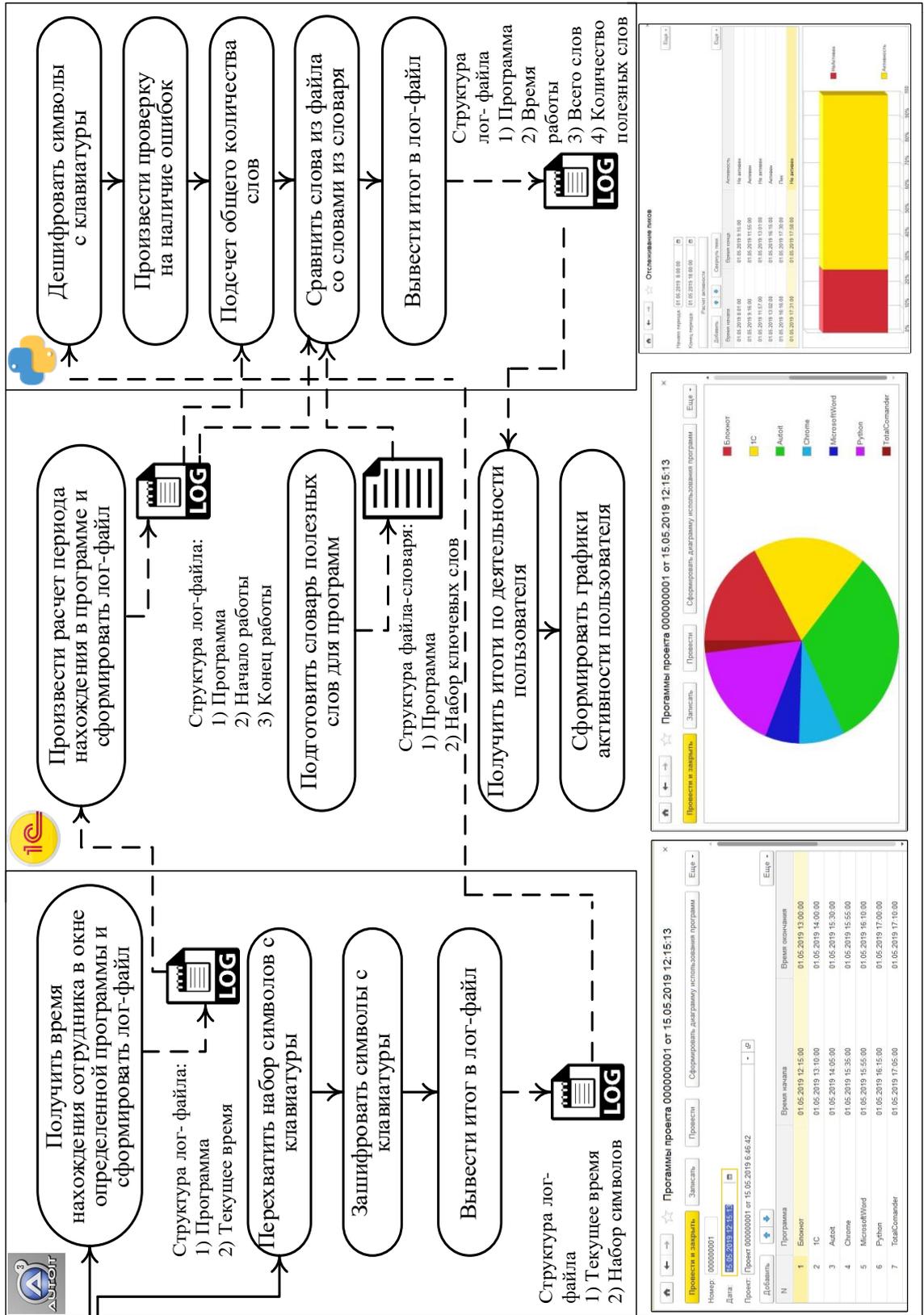


Рисунок 2 – Концептуальная схема инструментария учета и анализа рабочего времени сотрудников

Структура второго лог-файла включает в себя фиксацию времени начала и конца работы определенной программы, формально структуру лог-файла можно представить множеством (2).

Итог работы данного этапа – создание двух лог-файлов. Первый описывает окно программы и время нахождения в ней. Второй фиксирует время работы, за которое пользователь ввел набор символов. Формально полученный результат можно описать множеством (3).

На втором этапе формируется словарь и производится расчет периода нахождения в программе. Словарь формируется в управляющей системе «1С:Предприятие» с помощью экспертной комиссии. В дальнейшем предполагается развитие системы и внедрение нейронной сети, которая будет автоматически пополнять словарь часто используемыми выражениями, характерными для работы с данным программным обеспечением. Формально структуру словаря описывает множество (1).

На третьем этапе происходит дешифровка, анализ полезности или смысловой нагрузки слов, проверка на орфографические ошибки с помощью скрипта, разработанного на языке высокого уровня *python*. Все данные, которые поступают из предыдущих модулей, обрабатываются и из них выделяется важная информация. Также необходимо учитывать возможность того, что пользователь может совершить ошибку в написании некоторых слов и в этом случае, если слово оказалось «полезным», оно должно учитываться, поэтому необходимо учитывать проверку на наличие орфографических ошибок. Для этого используется библиотека *enchant*. После того, как была произведена проверка на ошибку, необходимо сравнить входит ли данное слово в словарь «полезных» слов для данной программы. Итогом работы данного этапа заключается формирование лог-файла, который включает себя время работы программы, общее количество слов и количество полезных слов. Формально данный лог-файл можно описать множеством (4).

Заключение

Предложен новый подход к автоматизированному анализу интервалов активности и простоя работы сотрудника за компьютером, что позволяет учитывать его полезное рабочее время при выполнении основных функциональных задач. Представлена формализованная постановка задачи, которая позволяет учитывать время на обдумывание решения пользователем и включать эти периоды подготовки в полезное рабочее время, т.к. даже если его активность взаимодействия с компьютером будет низкая – это не означает, что он не работает над поставленной задачей.

Для практической реализации данного подхода использовался стек трёх технологий: *AutoIt*, *Python*, платформа «1С:Предприятие». С помощью

языка *AutoIt* реализован перехват действий пользователя и запись в лог-файлы. Средствами языка *Python* осуществлялся анализ данных, отслеживались пики активности. Пользовательский интерфейс и общее управление программой реализовано на платформе «1С:Предприятие». Решение было представлено авторами на открытом чемпионате «IT-решения и инновации для бизнеса» *bizHack.Siberia* (г.Новосибирск, Академгородок, май 2019г.). Проект стал победителем в номинации «За оригинальность решения».

ЛИТЕРАТУРА

1. Программа Kickidler [Электронный ресурс] // www.kickidler.com – Режим доступа: <https://www.kickidler.com/ru/time-tracking.html>, свободный.– Загл. с экрана (дата обращения: 19.06.2019).
2. Программа StaffCop [Электронный ресурс] // www.staffcop.ru – Режим доступа: <https://www.staffcop.ru/efficiency>, свободный.– Загл. с экрана (дата обращения: 19.06.2019).
3. Программа ManicTime [Электронный ресурс] // www.manictime.com – Режим доступа: <https://www.manictime.com/teams/>, свободный.– Загл. с экрана (дата обращения: 19.06.2019).
4. Программа SkypeTime [Электронный ресурс] // skypetime.ru – Режим доступа: <https://skypetime.ru/chto-daet-ispolzovanie-skypetime/>, свободный.– Загл. с экрана (дата обращения: 19.06.2019).
5. Программа Yaware.TimeTracker [Электронный ресурс] // yaware.ru Режим доступа: <https://yaware.ru/what-is-yaware/for-business/>, свободный.– Загл. с экрана (дата обращения: 19.06.2019).
6. Программа Crocotime [Электронный ресурс] // crocotime.com – Режим доступа: <https://crocotime.com/ru/product/>, свободный.– Загл. с экрана (дата обращения: 19.06.2019).

S.N. Shirobokova, A.A. Katsuphev,

D.M. Zhevakin, O.N. Serikov, S.P. Vorobyev

ON THE APPROACH TO THE IMPLEMENTATION OF INTELLIGENT TOOLS FOR AUTOMATED RECORDING AND ANALYSIS OF STAFF TIME

*Platov South-Russian State Polytechnic University (NPI),
Novocherkassk, Russia*

The task of correct accounting of working time of the employee is closely related to the control of labor discipline and the use of working time, the evaluation of deadlines, accounting for the time spent on a particular task (which is especially important when paying by the hour, for example, when outsourcing services). The authors propose a new approach to the automated analysis of activity intervals and downtime of the employee at the computer. The article presents

a formalized formulation of the problem, allowing to take into account the periods of activity and potential inactivity in the preparation of useful working time. The aspects of practical implementation of software tools based on this approach are described. The stack of three technologies was used: AutoIt, Python, 1C:Enterprise platform. With the help of the script language AutoIt implemented interception of user actions and recording in the log files of the necessary information in encrypted form. Data analysis was carried out in Python, user activity peaks were monitored. On the platform "1C:Enterprise" the user interface was implemented and the General management of the program was carried out.

Keywords: employee working time accounting, formalized task model, automated control of working time usage, AutoIt, Python, 1C:Enterprise platform.

REFERENCES

1. Kickidler program [Electronic resource] // www.kickidler.com – access Mode: <https://www.kickidler.com/ru/time-tracking.html>, free. The title from the screen (date: 19.06.2019).
2. StaffCop program [Electronic resource] // www.staffcop.ru – access Mode: <https://www.staffcop.ru/efficiency>, free. The title from the screen (date: 19.06.2019).
3. ManicTime program [Electronic resource] // www.manictime.com – access Mode: <https://www.manictime.com/teams/>, free. The title from the screen (date: 19.06.2019).
4. SkypeTime program [Electronic resource] // skypetime.ru – access Mode: <https://skypetime.ru/chto-daet-ispolzovanie-skypetime/>, free. The title from the screen (date: 19.06.2019).
5. Yaware.TimeTrackerprogram [Electronic resource] // yaware.ru access Mode: <https://yaware.ru/what-is-yaware/for-business/>, free. The title from the screen (date: 19.06.2019).
6. Crocotime program [Electronic resource] // crocotime.com – access Mode: <https://crocotime.com/ru/product/>, free. The title from the screen (date: 19.06.2019).