

Таким образом, системы (19) и (20) с начальным условием (21) имеют одно и то же решение, представляющее собой матрицу размера n^2 . Элементами ее являются интересующие нас вероятности перехода $p_{ij}(t)$, то есть вероятности перехода из состояния E_i в E_j за времени t .

Для решения задачи прогнозирования количества мигрантов в миграционном потоке, находящихся в состоянии E_i , фиксируем некоторый начальный момент $t = 0$, в который должно быть известно число мигрантов, находящихся в каждом возможном состоянии E_i ($i = 1, 2, \dots, n$). Обозначим это число через $N(0)$.

Составим и решим одну из систем: (19) или (20). Следует отметить, что если число состояний невелико, то для решения системы (19) или (20) можно воспользоваться формулами (22) и (23). Однако при большом числе состояний операции по вычислению соответствующих матричных рядов нужно производить с помощью современных технических средств.

Полученное решение системы дифференциальных уравнений марковской цепи, то есть величины $p_{ij}(t)$ ($i = 1, 2, \dots, n$), умножим исходное число мигрантов, находящихся в состоянии E_j , то есть число $N_j(0)$, на $p_{ij}(t)$. В результате получим число мигрантов, находящихся в состоянии E_j , в которые перейдут за время t мигранты состояния E_i .

При большом числе состояний E_1, E_2, \dots, E_n решения системы дифференциальных уравнений высокого порядка связаны с большим объемом вычислений. Так при $n = 25$ необходимо решать систему из 625 уравнений. Однако, поскольку случайный процесс, анализируемый нами, является марковским, возникшие затруднения легко преодолеть.

На основании вышеуказанной теоремы, для достаточно больших значений t можно считать, что $p_{ij}(t)$ постоянны, то есть $p_{ij}(t) = p_{ij}$, где

p_{ij} – стационарные вероятности перехода. Следовательно, $p_{ij}(t) = 0$ и, таким образом, система дифференциальных уравнений марковской цепи равносильна системе алгебраических уравнений:

$$\sum_{j \neq k} p_{ij} q_{ik} = p_{ik} q_k, \quad (i = 1, 2, \dots, n). \quad (24)$$

Решением (24) являются постоянные величины. Применяя алгоритм прогнозирования, описанный выше появляется возможность определить численность миграционного потока, находящегося в состояниях E_i ($i = 1, 2, \dots, n$).

Заключение

Полученный в работе результат положен в основу методики отыскания коэффициентов стационарной марковской переходной матричной функции, что позволит строить краткосрочные и долгосрочные прогнозы при исследовании механического движения населения. На основе алгоритма прогнозирования, описанного выше, предложена имитационная модель, реализованная в среде C++Builder 6.

Предлагаемая методика отыскания вероятностей перехода может найти свое применение и в других предметных областях.

Литература

1. Оникиенко В.В., Поповкин В.А. Компьютерное исследование миграционных процессов. М.: Статистика, 1973. – 258 с.
2. Дуб Дж.Л. Вероятностные процессы. М., 1956. – 356 с.
3. Анго А. Математика для электро- и радиотехников. М.: Наука, 1965. – 272 с.
4. Моделирование социальных процессов. Под ред. Э.П. Андреева, Ю.Н. Габрилец. М.: Наука, 1970. – 317 с.

УДК 331.101.6:004.9

СИСТЕМА МОНИТОРИНГА В МАСШТАБЕ ПРЕДПРИЯТИЯ РАБОЧЕГО ВРЕМЕНИ СОТРУДНИКОВ, АКТИВНО ИСПОЛЬЗУЮЩИХ КОМПЬЮТЕР

Бочкин А.В., Федосин С.А.

Рассматривается возможность повышения производительности труда работающих за компьютером сотрудников в масштабе предприятия. Предложена информационная система, позволяющая существенно автоматизировать расчет производительности сотрудников для конкретных задач с целью улучшения организации рабочего времени.

Введение

За последние годы область, называемая поддержкой принятия управленческих решений, в результате быстрого развития компьютерных технологий стала приобретать новые очертания. Все большую актуальность стала приобретать задача

автоматизированного контроля эффективности использования рабочего времени (РВ) персоналом.

Масштабное внедрение компьютеров в работу предприятий создало предпосылки для создания специального программного обеспечения, которое позволило бы автоматизировать мониторинг РВ сотрудников, основываясь на использовании сотрудниками их основного рабочего инструмента - компьютера.

Основными задачами, возложенными на это программное обеспечение, являются:

- предоставление точной и своевременной информации о том, на что уходит РВ сотрудников за компьютером.

- прогнозирование затрат времени на выполнение проектов на основе накопленной в системе статистической информации.

Задача развертывания программного комплекса в масштабе предприятия требует системного подхода к построению архитектуры.

Постановка основных целей и задач

Рассмотрим, как сейчас решается проблема учета РВ. Перечень существующих методологий учета РВ и их сравнительный анализ приведен в таблице 1.

Очевидно, что периоды РВ различных составляющих могут чередоваться большое количество раз и в различной последовательности. Вести сбор и анализ такой информации без специальных средств автоматизации не представляется возможным. Следует отметить, что все существующие системы управления ERP, ERP II, CRM, CPM, EPM, BPM [2; 5] не используют методы учета РВ, отличные от ведения табеля учета РВ, что существенно снижает эффективность их применения [1; 3-4].

Таблица 1. Сравнительные характеристики методологий учета РВ

Составляющие РВ	Ручной табель РВ	Автоматический табель РВ	Рассматриваемая система
Время не нахождения на рабочем месте	+	+	+
Время бездействия при нахождении на рабочем месте			+
Время на решение задач, не связанных с интересами предприятия			+
Время на решение задач, определяемых руководством			+

В настоящее время все большее развитие получил проектный подход управления в силу своей высокой эффективности. Управление проектами (project management) — область деятельности, в ходе которой определяются и достигаются определенные цели, а также оптимизируется использование ресурсов (таких как время, деньги, труд, материалы, энергия, пространство и др.) в рамках некоторого проекта [6]. Рассматриваемая система позволяет прогнозировать затраты времени на выполнение проектов на основе метрики затрат времени на работу с программным обеспечением.

Предположим, что в результате эксплуатации системы накопились данные о затратах времени на решение типовых задач (см. таблицу 2).

Таблица 2. Матрица удельных затрат времени на задачи по сотрудникам

Сотрудники	Задача 1	Задача i	Задача n
Сотрудник 1	$e(1,1)$	$e(1,i)$	$e(1,n)$
Сотрудник k	$e(k,1)$	$e(k,i)$	$e(k,n)$
Сотрудник m	$e(m,1)$	$e(m,i)$	$e(m,n)$

Здесь $e_{ki} = \sum_{j=1}^p t_{kij} + r_{ki}$, где r_{ki} — дополнительное некомпьютерное время, t_{kij} — удельные затраты времени на программу j по задаче i для сотрудника k (см. таблицу 3).

Таблица 3. Матрица удельных затрат времени на задачи по программам для сотрудника k

Задачи	Группа программ 1	Группа программ j	Группа программ p
Задача 1	$t(k,1,1)$	$t(k,1,j)$	$t(k,1,p)$
Задача i	$t(k,i,1)$	$t(k,i,j)$	$t(k,i,p)$
Задача n	$t(k,n,1)$	$t(k,n,j)$	$t(k,n,p)$

Объем работы для проекта X составит $Q_x = \sum_{i=1}^n Z_i$, где Z_i — объем работ для задачи i , который распределяется между сотрудниками так, что $Z_i = \sum_{k=1}^m e_{ki} \times z_{ki}$. Коэффициенты z_{ki} определяются для проекта X как показано в таблице 4.

Таблица 4. Матрица коэффициентов для проекта X

Сотрудники	Задача 1	Задача i	Задача n
Сотрудник 1	$z(1,1)$	$z(1,i)$	$z(1,n)$
Сотрудник k	$z(k,1)$	$z(k,i)$	$z(k,n)$
Сотрудник m	$z(m,1)$	$z(m,i)$	$z(m,n)$

Время выполнения проекта X определяется следующим образом

$$T_x = T_s + \sum_{k=1}^m \sum_{i=1}^n z_{ki} \times \sum_{j=1}^n t_{kij} + r_{ki}.$$

Коэффициенты z_{ki} выбираются так, чтобы T_x было минимальным. При этом должны учитываться дополнительные ограничения на график рабочего времени сотрудников, приоритетность задач и другие ограничения. Поэтому вводятся дополнительные затраты времени на синхронизацию графика работ. Величины и могут рассчитываться при помощи различных систем планирования сроков выполнения проектов, таких Microsoft Project или других подобных систем.

Таким образом, рассматриваемая система служит следующим целям:

- определение оптимального объема загрузки сотрудников работой для обеспечения персонала работой ровно настолько, насколько он может ее выполнить в рамках своего РВ;

- предотвращение убытков организации из-за потерь РВ;

- прогнозирование затрат времени на выполнение проектов на основе накопленной в системе статистической информации и метрики затрат времени.

Для создания такой системы на практике предстоит решить следующие проблемы:

- учет времени работы программного обеспечения.

- учет времени активной работы персонала с программным обеспечением.

- классификация работы персонала в зависимости от используемого ПО.

- учет работы структурных подразделений предприятия.

- учет времени, затраченного сотрудниками с пользой для предприятия.

- разбиение затрат времени по задачам для оптимизации управления и прогнозирования.

- задание планируемого графика РВ и последующего сравнения реальным графиком.

- учет времени, проведенного персоналом на работе.

- формирование метрики расчета затрат времени на работу с программным обеспечением.

Архитектура программного комплекса

Архитектура комплекса должна быть основана на том, что система функционирует в масштабе предприятия, которое имеет иерархическую структуру, в которой можно выделить различные отделы и подразделения.

Исходя из требований к надежности системы, независимости от центрального сервера реализована агентская архитектура. Агент – модуль, собирающий данные с компьютеров сотрудников. Обзоратель – модуль управления агентами и формирования отчетов.

Модули агентов устанавливаются на компьютерах, по которым планируется вести учет РВ. На компьютерах, которые планируется использовать для просмотра отчетов, устанавливаются модули обзорателей.

Для того чтобы обеспечить разграничение доступа к агентам вводится политика учетных записей. Таким образом, для каждого агента существует свой набор учетных записей, для которых разрешен доступ к этому агенту. При этом возможно настроить определенные обзоратели для доступа к определенным агентам. Например, руководитель конкретного отдела может иметь доступ только к компьютерам своего отдела, а руководитель всей организации будет иметь доступ ко всем компьютерам всех отделов.

Преимуществами агентской архитектуры являются:

- возможность быстрых децентрализованных внедрений, способных к последующему объединению.

- независимость от наличия и работоспособности центрального сервера и наличия соединения с ним.

Возможная схема взаимодействия представлена на рис. 1.

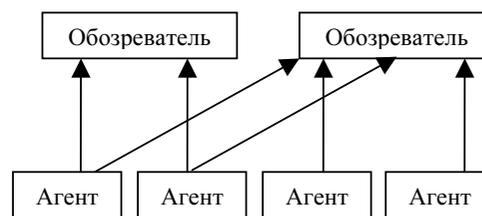


Рис. 1. Пример организации схемы взаимодействия

При практической реализации системы разработаны блоки алгоритмов:

- алгоритмы сбора и обработки данных;
- алгоритмы сетевого развертывания и функционирования программного комплекса в масштабе предприятия;
- вспомогательные алгоритмы.

Предоставление отчетов в системе базируется на использовании следующих алгоритмов работы с временными цепочками.

1. Установка интервалов в цепочках и их изъятие.
2. Пересечение временных цепочек.
3. Слияние временных цепочек с учетом графика.
4. Преобразование временных цепочек в данные по часам и дням.

Алгоритмы работы с временными цепочками

Временные цепочки – это последовательность интервалов времени. Для более быстрой навигации по интервалам используется индекс дней. Элемент массива дней содержит в себе смещение элемента интервала в массиве интервалов, начального для данного дня, а также высчитанное по интервалам суммированное время для данного дня. Один интервал может перекрывать несколько дней и быть начальным для нескольких разных дней. Массив дней создан для обеспечения возможности быстрой работы с массивом интервалов, а также для быстрого вычисления статистики, сгруппированной по дням, и общей суммарной статистики. Такая структура создана для обеспечения возможности группировки процессов в группы по направлениям и предоставления точной статистики в случае перекрытия времени процессов. Также это позволяет выдавать информацию о точных моментах запуска и останова процессов. Обработка данных реализована с учетом высоких требований к производительности и содержит в себе огромное количество различных вариантов проработки условий. Для обеспечения возможности группировки предусмотрены процедуры объединения массивов интервалов, что влечет за собой также частичную поправку массива дней. Объединение также необходимо в случае запуска нескольких экземпляров одного процесса, а также при расчете интервалов активности.

Если упростить ситуацию, то можно считать временной цепочкой только совокупность интервалов времени, опустив индекс дней. Тогда временную цепочку можно представить в следующем виде $X = \bigcup_{i=1}^n x_i$, где x_i – интервальное множество. Интервальное множество – множество замкнутого интервала $a = [a_1, a_2]$, которое рассматривается как множество заключенных в нем временных отсчетов.

Алгоритмы установки интервалов в цепочках и их изъятия

Базовой функцией для работы с временными цепочками является функция добавления отде-

льного интервала во временную цепочку. Опишем ее алгоритм.

На входе функция получает временную цепочку, время начала и окончания интервала времени для добавления. Результатом работы функции является объединение существующей временной цепочки с заданным интервалом времени:

$$z = X \cup y,$$

где X – временная цепочка, y – интервальное множество. На рис. 2 показано как при добавлении в цепочку 1 интервала получается цепочка 2.

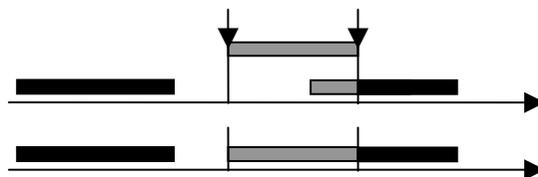


Рис. 2. Добавление интервала в цепочку

Одной из типовых задач при расчете статистики является выделение из временной цепочки ее части, взятой за определенный интервал времени. Это необходимо, например, в случае, когда руководителю нужен отчет не за весь интервал времени, а за какой-либо меньший временной интервал:

$$Z = F(X, y),$$

где X – временная цепочка, y – интервальное множество.

На входе функция получает исходную временную цепочку, границы интервала времени (t_1, t_2), информацию за который необходимо получить из исходной временной цепочки. На выходе получается временная цепочка за требуемый интервал времени, как показано на рис. 3.

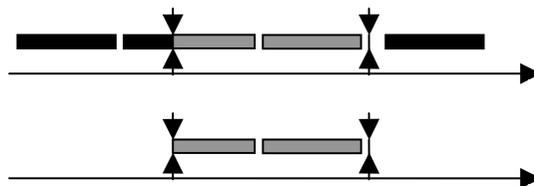


Рис. 3. Изъятие части цепочки

Алгоритмы пересечения временных цепочек

Разрабатываемая система должна решать задачу задания планируемого графика РВ и последующего сравнения реальным графиком. Это означает возможность задавать расписание РВ с

возможностью учета только РВ (исключая пере-
рывы на обед, ночное время и т.д.). Очевидно, что
рациональным является представить график РВ в
виде временных цепочек. В этом случае для того
чтобы учесть только необходимые интервалы
времени в соответствии с графиком необходимо
осуществить выполнение операции пересечения
этих временных цепочек. В результирующей вре-
менной цепочке окажутся только общие для двух
исходных временных цепочек интервалы:

$$Z = X \cap Y,$$

где X, Y – временные цепочки. Как показано на
рис. 4, при выполнении операции пересечения
временных цепочек 1 и 2 мы получаем цепочку 3.



Рис. 4. Пересечение временных цепочек

Алгоритмы слияния временных цепочек с учетом графика

В операционной системе всегда одновременно
выполняется большое количество процессов, и
может получиться так, что в системе может быть
запущено одновременно несколько программ,
относящихся к одной группе программного обес-
печения (например, офисные, системные, разра-
ботка и т.д.). На следующей диаграмме Ганта [4]
показано, как может происходить распределение
процессов во времени, рис 5. По вертикали отло-
жены программы, по горизонтали – время.

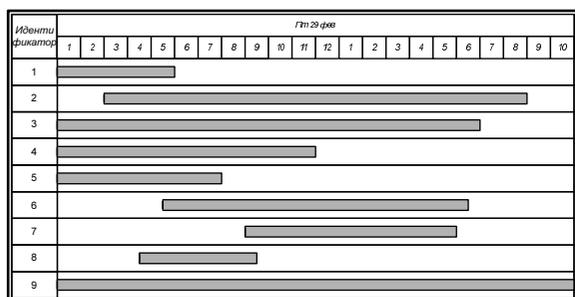


Рис. 5. Распределение процессов во времени

Руководителю необходимо предоставить
отчет о том, сколько времени были запущены
программы некоторых групп за определенный
промежуток времени. Например, он может за-

просить, сколько времени работали программы
группы «Разработка» за день. Простое сложение
времени не даст удовлетворительного ответа, так
как при этом может получиться, что время рабо-
ты окажется больше чем запрашиваемый интер-
вал времени, а это неправильно. Для разрешения
этой проблемы необходимо вычислять пересече-
ние (объединение) временных цепочек. При этом
если программа А работала с $t1$ до $t2$, и програм-
ма Б работала также, то результирующее время
будет равно $t2-t1$, а не $(t2-t1) * 2$ как в варианте
простого сложения. Слияние временных цепочек
необходимо проводить с учетом графика РВ, по-
этому с результатом слияния необходимо провес-
ти операцию пересечения:

$$Z = (X \cup Y) \cap S,$$

где X, Y, S – временные цепочки.

На рис. 6 показано как происходит слияние це-
почек 1 и 2 с учетом графика РВ 3. Результатом
является цепочка 4.

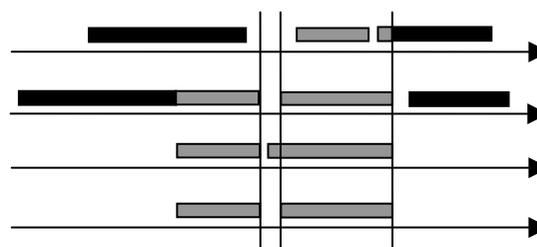


Рис. 6. Слияние временных цепочек с учетом графика

Алгоритмы преобразования временных цепочек в данные по часам и дням

Очевидно, что предоставлять руководителю
необходимо в виде понятных для осмысления от-
четов, а не в виде временных цепочек. Поэтому
необходимо реализовать функционал по преобро-
зованию временных цепочек в сгруппированную
по часам и дням информацию:

$$Z = F(X), \text{ где } X \text{ – временная цепочка.}$$

Таблица 5. Таблица по часам

Час	1	2	3	4	5	6	...	23	24
Время	t1	t2	t3	t4	t5	t6	...	t23	t24

Таблица 6. Таблица по дням

День	1	2	3	4	5	6	...	30	31
Время	t1	t2	t3	t4	t5	t6	...	t30	t31

Таким образом, из временной цепочки получаются таблицы с разбиением по часам и с разбиением по дням. Эта информация является наглядной иллюстрацией использования РВ офисными сотрудниками. На рис. 7 показан один из возможных вариантов отчетов, сгенерированных системой.

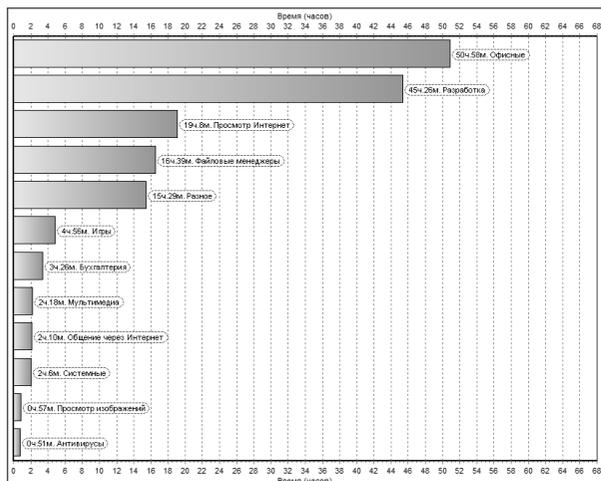


Рис. 7. Отчет в виде столбиковой диаграммы

Заключение

Созданная система предназначена для повышения в масштабе предприятия производительности труда сотрудников, чья деятельность связана с использованием компьютеров.

Система имеет более 40 внедрений в России, Украине, Казахстане, Беларуси, что однозначно

подтверждает ее актуальность на практике. Дополнительная информация о системе и ее trial-версия доступны на сайте www.maxapt.ru.

Литература

1. Бонни Бьяфоре. Все по плану! Успешное управление проектами с использованием Microsoft Project = On Time! On Track! On Target!: Managing Your Projects Successfully with Microsoft Project. М.: Microsoft Press, 2006. – 304 с.
2. Духонин Е.Ю., Исаев Д.В., Мостовой Е.Л. Управление эффективностью бизнеса. Концепция Business Performance Management. Под ред. Г.В. Генса. М.: Альпина Бизнес Букс, 2005. – 269 с.
3. Шлямин К.В. Автоматизация управленческого учета в финансово-промышленной группе: особенности, этапы реализации, результаты // Банки и технологии. № 3, 2005. – 200 с.
4. Управление проектами: Основы профессиональных знаний. Национальные требования к компетенции специалистов / Под ред. В.И. Воропаева. М.: СОВНЕТ, «Кубс Групп», 2001. – 265 с.
5. РБК // Аналитический обзор рынка систем класса СРМ (BPM) на III квартал, 2005. – 41 с.
6. Управление проектами – Википедия [Электронный ресурс] // Режим доступа: <http://ru.wikipedia.org>.

Поздравляем!

Всех читателей, авторов и деловых партнеров журнала ИКТ
с Наступающим 2009 годом!

Желаем здоровья, счастья и новых успехов в науке и
производственной деятельности.

Редакционная коллегия журнала ИКТ