

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

Государственное образовательное учреждение
высшего профессионального образования

«САНКТ-ПЕТЕРБУРГСКАЯ ГОСУДАРСТВЕННАЯ ЛЕСОТЕХНИЧЕСКАЯ
АКАДЕМИЯ имени С. М. Кирова»

Кафедра информатики и информационных систем

ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ: ТЕОРИЯ И ПРАКТИКА

Сборник научных трудов

Выпуск 2

Санкт-Петербург
2009

Рассмотрен и рекомендован к изданию
Ученым советом факультета экономики и управления
Санкт-Петербургской государственной лесотехнической академии
4 июня 2009 г.

Редакционная коллегия:

А. М. Заяц, кандидат технических наук, профессор (отв. редактор),
И. В. Ганичев, кандидат технических наук, доцент (отв. секретарь),
И. В. Панфилов, доктор технических наук, профессор

Составитель

И. В. Ганичев, кандидат технических наук, доцент

Рецензент

доктор технических наук, профессор **В. И. Николаев**
(Санкт-Петербургский филиал ОАО
«Российские телекоммуникационные сети»)

УДК 630

Информационные системы и технологии: теория и практика: сб.
науч. тр. Вып. 2. / отв. ред. А. М. Заяц. — СПб.: СПбГЛТА, 2009. — 128 с.

ISBN 978-5-9239-0163-4

Сборник подготовлен по материалам кафедры вуза, представленным на научно-технической конференции факультета экономики и управления СПбГЛТА в феврале 2009 года, и практических работ, выполненных ее сотрудниками.

Темплан 2009 г. Изд. № 217.
ISBN 978-5-9239-0163-4

© Санкт-Петербургская государственная
лесотехническая академия (СПбГЛТА), 2009

А.М. Заяц, кандидат технических наук, профессор

ИТОГИ НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЫ

Коллектив кафедры информатики и информационных систем в юбилейном 2008 году проводил исследования, ориентированные на разработку научно-методических трудов, которые направлены на применение современных информационно-образовательных технологий в учебном процессе, таких как:

- 1) использование компьютерных технологий и математических пакетов в решении оптимизационных задач лесопромышленного комплекса;
- 2) применение мультимедийных, Internet\Intranet и web-технологий в теоретической и практической подготовке студентов;
- 3) совершенствование методики преподавания учебных дисциплин направления 230200 «Информационные системы».

По итогам НИР за 2008 г. коллектив кафедры в системе автоматизированной рейтинговой оценки ЛТА занял 17 место из 49.

В рамках плана мероприятий, приуроченных к *40-летию кафедры*, была организована и проведена юбилейная научно-практическая конференция «*Информационные системы и технологии: теория и практика*», по результатам которой был выпущен сборник трудов, в котором 18 научных работ представили преподаватели кафедры:

1. *Заяц А.М.* Новый подход в практической подготовке студентов лесотехнической академии.
2. *Богатырев В.А., Богатырев С.В.* Векторная оптимизация структуры кластера.
3. *Панфилов И.В.* Проблемное обучение.
4. *Терентьев И.В., Неверовский В.Ю., Шубина М.А.* Алгоритмы автоматического определения возраста растений и животных по фотографиям.
5. *Терентьев И. В., Шубина М. А, Морус Г. А.* Усовершенствованный алгоритм классификации лесохозяйственных объектов на аэрокосмических изображениях земной поверхности.
6. *Уткин Л.В., Нгуен Ван Х.* Пессимистический подход к многокритериальному принятию решений в рамках теории случайных множеств.
7. *Васильев Н.П., Ковешников А.И., Пресняков В.А.* Вариант решения задачи экспорта данных.
8. *Горбачев В.А., Потокин А.Ф.* Создание базы данных электронного каталога коллекции «Гербария им. Бородина» СПб ГЛТА.

9. *Курилова М. Н.* Опыт проведения и участия в олимпиадах по информатике и программированию кафедры информатики и информационных систем.

10. *Лебедев М.О.* Принципы создания динамически настраиваемых информационных систем.

11. *Лушкин Н.В.* Управление процессом противоточного выщелачивания пульсациями растворителя.

12. *Михайлов АА.* Информационные технологии – организатор и катализатор образовательного процесса.

13. *Симанова Н.В.* Групповое многокритериальное принятие решений на основе алгебры предпочтений.

14. *Хабаров С.П., Колмогорцев Е.Л.* К вопросу об оценке времени реакции распределенной системы управления с резервированием сетевой коммуникационной подсистемы.

15. *Халиков М.И.* Использование механизма бизнес-процессов для автоматизации учетных задач в программе «1С: Предприятие 8.0».

16. *Ярин Б.Д.* Электронные учебные пособия.

Две работы опубликованы преподавателями кафедры совместно со студентами:

1. *Ганичев И.В., Троицкая Е.Г., Троицкая А.Г.* Системные исследования иерархических моделей ЭВМ.

2. *Заяц А.М., Гоголевский А.С.* Автоматизация разработки лесохозяйственных регламентов и проектов освоения лесов на платформе виртуальных лесничеств ЭВМ.

В течение 2008 г. преподаватели кафедры подготовили и выпустили следующие учебные пособия и методические рекомендации:

1. *Панфилов И.В., Панфилова Е.И., Заяц А.М.* Проблемы защиты информации в компьютерных сетях и системах: учебное пособие. СПб.: СПбГЛТА, 2008.

2. *Лебедев М.О.* Технология программирования: учебное пособие. СПб.: СПбГЛТА, 2008.

3. *Васильев Н.П., Шилкина М.Л.* Технология публикации расчетов web-обозревателей: учебное пособие. СПб.: СПбГЛТА, 2008.

4. *Заяц А.М.* Информационные системы: методические рекомендации по курсовому проектированию «Проектирование информационных систем». СПб.: СПбГЛТА, 2008.

В российских журналах и сборниках трудов преподавателями опубликованы следующие работы:

1. *Заяц А.М.* Виртуальное лесничество подготовит грамотные кадры // ЛЕСПРОМинформ. 2008. № 2. С. 26–28.

2. Заяц А.М. Учебные виртуальные лесничества в системе заочного обучения студентов лесотехнической академии // Тр. СПбГЛТА. Актуальные проблемы высшей школы. – СПб.: СПбГЛТА, 2008. Т. 0. С. 152–155.

3. Заяц А.М. Экспертно-консалтинговая информационно-образовательная платформа в лесном образовании и бизнесе // V Международная науч.-техн. конф. «Актуальные проблемы лесного комплекса». – ВологГТУ, 2008. С. 1–6.

4. Заяц А.М., Алексенко А.Н., Цветкова О.А. Парк лесотехнической академии как информационно-образовательный ресурс в системе обучения студентов // Тр. СПбГЛТА. Актуальные проблемы высшей школы. – СПб.: СПбГЛТА, 2008. Т. 0. С. 155–157.

Доклады на российских конференциях и форумах:

1. Заяц А.М. Учебные виртуальные лесничества в практической подготовке студентов ЛТА // Лесной саммит России «ИНТЕРЛЕС–ЛИСИНО-2008».

2. Заяц А.М. Виртуальная виртуозность «Леса России».

3. Заяц А.М. Экспертно-консалтинговая информационно-образовательная платформа // Международный лесной форум 2008.

Завершены исследования и представлены материалы в итоговый отчет в рамках гранта Рособразования РНП 2.2.2.2.7182 «Разработка научно-образовательной программы по устойчивому развитию лесного комплекса России на базе международного и образовательного сотрудничества».

А.М. Заяц, кандидат технических наук, профессор
Б.В. Савельев, А.С. Гоголевский, студенты

СИСТЕМА МНОГОПОЛЬЗОВАТЕЛЬСКОГО ДОСТУПА К РЕСУРСАМ УЧЕБНЫХ ВИРТУАЛЬНЫХ ЛЕСНИЧЕСТВ

Реализация требований по информатизации образовательного процесса на кафедре информатики и информационных систем осуществляется в рамках учебных виртуальных лесничеств.

Учебные виртуальные лесничества представляет собой информационно-образовательную систему с соответствующими базами данных и приложений, объединенных общей телекоммуникационной программно-аппаратной средой обработки и обмена информацией и функционирующих в соответствии с определенными технологическими, образовательными и правовыми нормами.

В такой системе одной из первоочередных является задача выбора и внедрения механизма поддержки распределенной обработки данных и обеспечения многопользовательского доступа к ресурсам системы, как по корпоративной сети академии, так и через Internet.

Данный механизм должен обеспечивать регистрацию пользователей, их авторизованный вход в систему и разграничение доступа, возможность групповой обработки документов с сохранение их образов.

Так как все учебные лесничества хранятся на сервере академии, то студенты (пользователи) «входят» в свое учебное виртуальное лесничество под собственным логином и паролем и могут вносить изменения и производить настройки только в программной среде своего УВЛ, в соответствии с технологией решения поставленной задачи.

Реализация этого механизма позволит поставить однозначное соответствие между конкретным пользователем и доступным ему множеством ресурсов для реализации функциональных задач в рамках своего учебного виртуального лесничества с учетом внутренних и внешних информационных и ресурсных связей, определенных заданием.

Визуализация информационной среды для пользователя осуществляется посредством графического интерфейса, через который каждый из них получает доступ к информационным ресурсам и приложениям УВЛ.

Разработка пользовательского интерфейса проводится с учетом следующих требований:

- графический интерфейс должен соответствовать текущим стандартам и одновременно быть «дружественным» с удобной системой навигации и поиска;

- найденная и представленная пользователю информация, снабженная гиперссылками и динамическими сносками должна соответствовать первоисточнику;
- графический интерфейс должен обеспечивать необходимость и достаточность вызываемых ресурсов для анализа и обработки информации;
- представляемые электронные формы документов должны соответствовать бумажным аналогам;
- процедуры заполнения и редактирования документов и отчетных форм должны быть понятными как для профессионала, так и для простого пользователя;
- процедуры поиска должны поддерживать разные форматы документов и выдачу их на различные устройства вывода с возможностью публикации в Интернет.

Интерфейс должен обеспечивать администратору системы возможность гибкой настройки режима распределенного выполнения функциональных процедур с использованием сервера приложений и web-сервера реализующего Internet/Intranet-технологии.

Разработанная таким образом система доступа позволит одинаково понимать и строить информационную модель, использовать одни и те же параметры для характеристики информационных объектов и ресурсов в интегрированной распределенной информационной системе УВЛ и решить задачу санкционированного доступа к ресурсам в едином информационном пространстве.

Библиографический список

1. Заяц А.М. Новый подход в практической подготовке студентов Лесотехнической академии//Сборник научных трудов "Информационные системы и технологии: теория и практика" -СПб.: СПбГЛТА, 2008, 4 - 13 с.
2. Заяц А.М. Виртуальное лесничество подготовит грамотные кадры//ЛЕСПРОМинформ. 2008. № 2, 26-28 с.
3. Заяц А.М. Учебные виртуальные лесничества в системе заочного обучения студентов лесотехнической академии//Труды Санкт-Петербургской Государственной Лесотехнической академии. Актуальные проблемы высшей школы – СПб.: СПбГЛТА, 2008. 152 - 155 с.
4. Заяц А.М., Гоголевский А.С. Автоматизация разработки лесохозяйственных регламентов и проектов освоения лесов на платформе виртуальных лесничеств //Сборник научных трудов "Информационные системы и технологии: теория и практика" – СПб.: СПбГЛТА, 2008, 13-19 с.

Л.В. Уткин, доктор технических наук, профессор
С.И. Затенко, аспирант

ИНТЕРВАЛЬНАЯ МОДЕЛЬ ТЕСТИРОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ НА ОСНОВЕ ОБОБЩЕННЫХ ПРОЦЕССОВ ПУАССОНА

Введение

Проявление ошибок в программном обеспечении (ПО) интенсивно изучалось в литературе в целях улучшения характеристик программ. Большое число моделей надежности программного обеспечения было разработано в последние десятилетия, но ни одна модель не дает точного прогноза для различных программ. Причина этого заключается в том, что в каждой модели предполагается наличие одного определенного распределения вероятностей таких случайных величин, как время до отказа ПО или число ошибок, выявленных за заданное время. Это не всегда соответствует реальной ситуации, когда данные тестирования статистически неоднородны и их количество ограничено. Детальный критический обзор известных вероятностных моделей надежности ПО приведен в работе [1].

Основной идеей моделирования надежности ПО в большинстве существующих моделей является попытка статистически описать процесс отладки или тестирования программы (исправления ошибок и тестирования до момента проявления очередной ошибки). Таким образом, для анализа надежности ПО с использованием статистических моделей, их параметры оцениваются на основе имеющихся данных об отказах.

В работе предлагается модель, основанная на процессах Пуассона, в соответствии с которыми имеющиеся исходные данные представляются в виде обнаруженного числа ошибок ПО (отказов) за определенное время. Предлагаемая модель будет называться интервальной моделью, так как основные показатели надежности ПО, вычисляемые с ее помощью, являются интервальными. Основная идея построения модели заключается в использовании обобщенного байесовского вывода [2], где вместо одного априорного распределения вероятностей рассматривается множество распределений. Это позволяет обойти сложный вопрос выбора подходящего априорного распределения вероятностей в ситуации, когда о программе ничего не известно перед тестированием.

Стандартные модели тестирования

Одним из инструментов для создания класса моделей тестирования ПО, исходными данными для которых являются значения числа отказов ПО, обозначаемые $N(t)$, за определенный период времени t , являются

процессы Пуассона, согласно которым вероятность числа отказов k за время t равна

$$p(k) = \Pr\{N(t) = k\} = \frac{(\lambda t)^k \exp(-\lambda t)}{k!}, \quad k = 0, 1, 2, \dots$$

Здесь λ – интенсивность отказов, которая оценивается при помощи метода максимума функции правдоподобия, используя в качестве исходных данных значения числа отказов в заданные интервалы тестирования и отладки.

Байесовский подход к анализу статистических данных

Если анализируемая случайная величина (например, число отказов за заданное время) имеет распределение вероятностей с вектором неизвестных параметров \mathbf{b} (в рассматриваемой модели тестирования $\mathbf{b} = (\lambda)$), то в соответствии с байесовским подходом [3] эти параметры рассматриваются, как случайные величины с априорной плотностью вероятностей $\pi(\mathbf{b}|\mathbf{c})$ с параметрами \mathbf{c} .

Центральным элементом байесовского подхода является вывод апостериорного распределения неизвестных параметров при получении статистических данных на основе теоремы Байеса. Пусть статистическая информация собрана в виде вектора $\mathbf{K} = (k_1, \dots, k_n)$ наблюдаемых значений случайной величины. Пусть $p(k)$ – вероятность того, что наблюдаемое значение числа отказов равно k . Тогда апостериорная плотность $\pi(\mathbf{b}|\mathbf{K}, \mathbf{c})$ вычисляется как

$$\pi(\mathbf{b}|\mathbf{K}, \mathbf{c}) \propto p(k_1) \cdots p(k_n) \cdot \pi(\mathbf{b}|\mathbf{c}).$$

Априорное распределение зачастую выбирается таким образом, чтобы упростить вычисления. Наиболее эффективный путь в этом направлении – это выбор *согласованных априорных распределений*. Если апостериорное распределение $\pi(\mathbf{b}|\mathbf{K}, \mathbf{c})$ и априорное распределение $\pi(\mathbf{b}|\mathbf{c})$ принадлежат одному семейству распределений, то π и p называются согласованными распределениями.

Если отбросить требование простоты вычислений, то предпочтительное априорное распределение должно минимально влиять на апостериорное распределение, а также учитывать отсутствие априорной информации о параметрах. Априорные распределения, моделирующие отсутствие априорной информации, называются *неинформативными*. Существует большое количество подходов для выбора неинформативного априорного распределения, имеющих свои достоинства и недостатки. Однако определим не одно априорное распределение, а целый класс P распределений π , для которого можно найти нижнюю и верхнюю вероятности события A как

$$\underline{P}(A) = \inf\{P_\pi(A) : \pi \in P\}, \quad \bar{P}(A) = \sup\{P_\pi(A) : \pi \in P\}.$$

Когда априорной информации почти нет, для этого класса $\underline{P}(A) \rightarrow 0$, а $\bar{P}(A) \rightarrow 1$. Это означает, что априори событие A может иметь любую вероятность.

Обобщенная модель отрицательного биномиального распределения

Если число отказов имеет распределение Пуассона, то априорное согласованное распределение для параметра λ , который рассматривается в качестве случайной величины, является гамма-распределением, обозначаемым $G(a, b)$ с параметрами $a > 0$, $b > 0$, т.е. $\lambda \square G(a, b)$.

Предположим, что уже было зафиксировано K отказов, наблюдаемых за период времени T . Тогда апостериорным распределением является $G(a + K, b + T)$. Отсюда прогнозируемое распределение k отказов за время t при условии, что суммарное число отказов K было зафиксировано за общее время T , имеет вид

$$P(k, t) = \int_0^{\infty} \frac{(\lambda t)^k \exp(-\lambda t)}{k!} \cdot G(a + K, b + T) d\lambda = \\ = \frac{\Gamma(a + K + k)}{\Gamma(a + K) k!} \left(\frac{b + T}{b + T + t} \right)^{\alpha + K} \left(\frac{t}{b + T + t} \right)^k.$$

Это выражение для отрицательного биномиального распределения.

Заменим в выражении для $P(k, t)$ параметры a и b параметрами α и s таким образом, что $a = s\alpha$ и $b = s(1 - \alpha)$. Тогда

$$P(k, t) = \frac{\Gamma(s\alpha + K + k)}{\Gamma(s\alpha + K) k!} \cdot \left(\frac{s(1 - \alpha) + T}{s(1 - \alpha) + T + t} \right)^{s\alpha + K} \left(\frac{t}{s(1 - \alpha) + T + t} \right)^k.$$

Следует отметить, что априорное математическое ожидание интенсивности отказов при условии, что $K = 0$ и $T = 0$, вычисляется в терминах новых параметров, как $E X = a / b = \alpha / (1 - \alpha)$. Дисперсия интенсивности отказов равна

$$\text{var}(X) = a / b + a / b^2 = \alpha / (1 - \alpha) + \alpha / s(1 - \alpha)^2.$$

Следовательно, изменяя параметр α в интервале от 0 до 1 можно получить интервал математического ожидания интенсивности отказов от 0 до ∞ . Аналогичное заключение можно сделать и для дисперсии.

Рассмотрим обобщенную модель, которая определяется как множество P всех отрицательных биномиальных распределений с фиксированным параметром s и множеством параметров $\alpha \in [0, 1]$. Параметр $s > 0$ определяет влияние априорного распределения на апостериорное. Он также определяет, как быстро нижняя и верхняя границы вероятностей событий сходятся по мере накопления статистических данных. Принимая множество распределений вместо одного распределения вероятностей, мы можем искать только нижнюю и верхнюю границы вероятностей событий

вместо точных значений для одного распределения. При этом нижняя и верхняя границы могут быть получены минимизацией и максимизацией этих вероятностей по всему множеству значений $\alpha \in [0, 1]$.

Так как вместо одного распределения в обобщенной модели используется множество всех распределений, то произвольная оценка $P(k, t)$ может быть получена как интервал с нижней и верхней границей. Заметим, что альтернативная форма записи функции распределения вероятностей числа отказов k за время t имеет следующий вид:

$$F(k, t) = 1 - \frac{B_q(k+1, r)}{B(k+1, r)}.$$

Здесь

$$r = s\alpha + K, \quad q = \frac{t}{s(1-\alpha) + T + t},$$

$B_q(M+1, r)$ - неполная бета-функция, определяемая как

$$B_q(k+1, r) = \int_0^q x^k (1-x)^{r-1} dx.$$

Если $q=1$, то неполная бета-функция совпадает с бета-функцией, т.е., $B_1(k+1, r) = B(k+1, r)$.

Из свойств бета-функции и неполной бета-функции следует, что функции распределения вероятностей убывает с увеличением параметра α . Здесь под убыванием функции распределения вероятностей понимается не зависимость функции от аргумента k , а убывание всех значений функции, т.е. $F_1(k, t) \geq F_2(k, t)$ если $\alpha_1 \leq \alpha_2$. Таким образом, максимальное значение функции распределения или верхняя граница \bar{F} достигается при $\alpha \rightarrow 0$ и равна

$$\bar{F}(k, t) = 1 - \frac{B_q(k+1, K)}{B(k+1, K)}, \quad q = \frac{t}{s + T + t}.$$

Минимальное значение функции распределения вероятностей достигается при $\alpha \rightarrow 1$ и равно

$$\underline{F}(k, t) = 1 - \frac{B_q(k+1, s+K)}{B(k+1, s+K)}, \quad q = \frac{t}{T + t}.$$

Таким образом, получены границы для функции распределения числа отказов за время t .

Отсюда нижняя $\underline{E}^{(s)}X$ (оптимистическая) граница ожидаемого числа отказов за время t после отладочного процесса определяется через известные выражения для математического ожидания случайной величины, имеющей отрицательное биномиальное распределение при $\alpha \rightarrow 0$, как

$$\underline{E}^{(s)}X = K \frac{t}{s + T}.$$

Фактически параметр s при вычислении нижней границы увеличивается

ет время тестирования на величину s . Верхняя $\bar{E}^{(s)} X$ (пессимистическая) граница ожидаемого числа отказов за время t после отладочного процесса определяется при $\alpha \rightarrow 1$ как

$$\bar{E}^{(s)} X = (s + K) \frac{t}{T}.$$

Параметр s при вычислении верхней границы увеличивает число отказов на величину s .

Заключение

В работе был предложен подход для получения границ математического ожидания числа отказов ПО или функции распределения вероятностей числа отказов за заданное время. Подход использует байесовское сужение множества возможных распределений Пуассона по мере поступления статистических данных тестирования. При этом параметр s позволяет управлять степенью осторожности получаемых результатов, т.е. делать соответствующие границы более или менее осторожными в зависимости от ответственности тестируемого ПО и последствий его отказов. Главной особенностью предлагаемого подхода является возможность получать результаты при любом объеме данных тестирования.

Библиографический список

1. Уткин Л.В., Шубинский И.Б. Нетрадиционные методы оценки надежности информационных систем. – СПб.: Любавич, 2000. – 173 с.
2. Уткин Л.В. Анализ риска и принятие решений при неполной информации. – СПб.: Наука, 2007. – 404 с.
3. Bernardo J.M., Smith A.F.M. Bayesian Theory. – Chichester: Wiley, 1994.

А.А.Никифоров, кандидат технических наук, доцент
Л.В.Уткин, доктор технических наук, профессор

АВТОМАТИЗИРОВАННАЯ СИСТЕМА РЕЙТИНГОВОЙ ОЦЕНКИ И МОНИТОРИНГА ОБРАЗОВАТЕЛЬНОЙ И НАУЧНОЙ ДЕЯТЕЛЬНОСТИ ВУЗА

В целях оценки деятельности профессорско-преподавательского состава и централизованного сбора данных, обработки и подготовки отчетов научно-исследовательской части и учебно-методического управления, а также для стимуляции и активизации творческой педагогической и научной деятельности сотрудников кафедр в течение календарного года, была разработана автоматизированная система рейтинговой оценки деятельности преподавателей, кафедр и факультетов. Система является основой для распределенного сбора и автоматизированного мониторинга аспектов научной и образовательной деятельности вуза.

В настоящее время, система апробирована и реализована для Санкт-Петербургской Государственной лесотехнической академии имени С.М.Кирова (СПб ГЛТА). Автоматизированная система рейтинговой оценки является неотъемлемой частью официального информационного портала СПбГЛТА. Получить доступ к системе могут только авторизованные пользователи через меню сайта: «Наука» >> «Автоматизированная система рейтинговой оценки» или по адресу <http://ftacademy.ru/rating/>.

При разработке системы применялся современный язык серверных сценариев PHP, а также язык скриптов - JavaScript. В качестве хранилища информации используется система управления базами данных MySQL, построенная по технологии клиент-сервер.

Доступ к системе осуществляется через модуль авторизации по логину и паролю, который выдается администратором портала персонально.

Выделены две категории пользователей:

1. Сотрудники кафедр;
2. Сотрудники администрации.

Соответственно, пользователи первой категории имеют доступ только к информации своей кафедры (рис. 1), а пользователи второй категории имеют доступ к информации всех кафедр, с возможностью корректировки и обработки информации.

Ввод данных сотрудниками кафедр организован по принципу самостоятельного заполнения полей форм, соответствующих определенной деятельности преподавателей.

Система охватывает только те виды деятельности преподавателя, которые демонстрируют его активность в учебном процессе и научно-

исследовательской работе, а также обеспечивает интегральный характер оценки. Система максимально приближена к аккредитационным показателям вузов.

Санкт-Петербургская Государственная Лесотехническая академия им. С. М. Кирова

Автоматизированная система рейтинговой оценки

Выход

Список научно-педагогических работников кафедры информатики и информационных систем

ФИО	Действие
Уткин Лев Владимирович	
Защитин Анатолий Моисеевич	
Джериков Вадим Вадимович	
Панфилов Иван Васильевич	
Халипов Мебин Исаакович	
Богатырев Владимир Анатольевич	
Горбачев Виктор Александрович	
Ганичев Игорь Викторович	
Киселева Светлана Владимировна	
Сымаганова Наталья Владимировна	
Ярин Борис Давыдович	
Васильев Николай Павлович	
Шубина Марина Александровна	
Михайлов Анатолий Аркадьевич	
Лебедев Михаил Олегович	
Геренштейн Игорь Вячеславович	
Кабаров Сергей Петрович	
Жижко Виталий Николаевич	
Шипилова Мария Львовна	
Луцкин Николай Васильевич	
Иванова Ирина Владимировна	
Смелова Наталья Борисовна	
Караганов Виктор Дмитриевич	

Добавить сотрудника

Добавление нового сотрудника в список научно-педагогических работников кафедры осуществляется кнопкой "Добавить сотрудника", которая вызывает соответствующую форму. В форме находится только основная информация: **Показатели достигнутой квалификации**.
 Для добавления сведений о **показателях активности в учебной работе** и **показателях активности в научной работе** необходимо выбрать соответствующего сотрудника и перейти в режим его редактирования .

В случае ошибочного ввода сотрудника, запись можно удалить .

Поля, которые отмечены *, требуют обязательного заполнения.
 Все сведения, которые Вы вводите, должны быть за **текущий календарный год**.

Внимание! Вводите только достоверную информацию, вся информация будет проверяться. **Некорректно введенные данные будут удалены и в расчет рейтинга не войдут.**

Рис. 1. Интерфейс системы, доступный сотрудникам кафедр.

Выделены три группы показателей:

1. Показатели достигнутой квалификации (такие как категория научно-педагогического работника; должность; звание; членство в советах и научных обществах);

2. Показатели активности в учебной работе (наличие учебников с грифом Минобрнауки РФ или УМО; учебных пособий с грифом УМО и без грифа; методических указаний, изданных вузовским тиражом; электронных учебных и учебно-методических пособий; разработанных и читаемых новых курсов лекций; курсов новых лабораторных работ (семинаров); разработанных и читаемых курсов лекций с компьютерными демонстрациями; создан-

ных новых программ по дисциплинам, зарегистрированным в методическом отделе; повышение квалификации за отчетный период);

3. Показатели активности в научной работе.

3.1. Диссертации (собственная защита кандидатской диссертации или докторской диссертации за отчетный период; защиты кандидатских или докторских диссертаций под Вашим руководством за отчетный период; оппонирование диссертаций и наличие отзывов на авторефераты диссертаций);

3.2. Публикации (статьи в зарубежных периодических журналах, в зарубежных сборниках, в центральных Российских журналах, в Российских сборниках и прочих изданиях; монографии; сборники научных трудов, в которых сотрудник кафедры является ответственным редактором, редактором, членом редколлегии, составителем; книги, написанные под редакцией);

Патентно-лицензионная работа (получено патентов на изобретения; подано заявок; положительных решений; поддерживаемых патентов; сделанных открытий; проданных лицензий; зарегистрированных программ для ЭВМ, баз данных, топологий интегральных микросхем);

Выставки и конференции (выставки, в которых сотрудник принимал участие; конференции, в которых сотрудник принимал участие без публикации тезисов; докладов (тезисов) на зарубежных и Российских конференциях);

Организация научно-исследовательской деятельности студентов (победители открытого конкурса на лучшую научную работу студентов, проводимые по приказу Минобробразования России, организованные факультетом; студенческие научные и научно-технические конференции; выставки студенческих работ, организованные факультетом; студенты, участвующие в НИР с оплатой труда; доклады студентов на научных конференциях; экспонаты, представленные на выставках с участием студентов; научные публикации студентов, изданные без соавторов - сотрудников академии за отчетный период; медали, дипломы, грамоты, премии и т.п., полученные студентами на конкурсах на лучшую НИР и на выставках; заявки, поданные студентами на объекты интеллектуальной собственности; студенческие проекты, участвовавшие в конкурсах грантов; гранты, выигранные студентами; стипендии Президента РФ, получаемые студентами);

Гранты, хоздоговорные работы (привлечение спонсорской помощи; наличие хоздоговорных работ; грантов РФФИ; зарубежных грантов; грантов Федеральных органов; подано заявок на гранты всех видов);

Показатели участия в информатизации СПб ГЛТА (официальный сайт, системы управления вузом).

Расчет рейтинговых показателей производится сотрудниками администрации после выполнения следующих операций:

- обработки введенной информации сотрудниками кафедр;
- определения весов показателей;
- проверки численности сотрудников кафедр.

После сверки информации и расчета рейтинговых показателей сотрудники администрации получают возможность формировать следующие отчетные материалы:

- Рейтинговые списки;
 - Факультетов;
 - Кафедр;
 - Сотрудников;
- Отчетные материалы;
 - по академии в целом;
 - по факультетам;
 - по кафедрам.

Система является открытой с точки зрения модификации и дополнения. Это позволяет безболезненно добавлять и удалять показатели, обеспечивая сбор данных наиболее полно. Структура базы данных представлена в виде таблиц связанных по идентификатору. Помимо основной таблицы, содержащей персональные сведения о сотрудниках, используются четыре таблицы для хранения следующей информации:

1. Учебники, методические пособия, монографии, а также статьи и тезисы докладов с их выходными данными.
2. Гранты и хоздоговорные работы.
3. Диссертации кандидатские и докторские.
4. Объекты, такие как выставки, конференции, экспонаты, лицензии.

Особенностью системы является возможность просматривать со страниц сайта информацию базы данных по категориям в виде форматированного списка с возможностью последующей печати.

Система используется различными службами и структурными подразделениями вуза в качестве информационной системы.

Управлением научных программ и проектов автоматизированная система применяется в качестве источника сводных данных по категориям и для формирования информационной базы для подготовки годовых отчетов академии о результатах учебной и научной деятельности и подготовки сведений для аттестации и аккредитации академии, сведений по основным показателям инновационной и научной деятельности для Комитета по науке и высшей школе Правительства Санкт-Петербурга, а также для подготовки к изданию «Перечня избранных публикаций сотрудников СПбГЛТА», тематических рубрик сборника «Известия Санкт-Петербургской ле-

сотехнической академии», включенного в перечень ВАК ведущих рецензируемых научных журналов и изданий.

Достоверность результатов анализа обосновывается тем, что преподаватели, наиболее активно занимающиеся наукой в академии, занимают первые позиции в рейтинговом списке. Кроме того, выполняется выборочный контроль правильности и достоверности предоставления данных, а также полный контроль «выбросов».

За 2006-2008 годы был накоплен определенный опыт реализации системы рейтинговой оценки, который, в свою очередь, показал ее жизнеспособность, важность и эффективность. С учетом накопленного опыта планируется дальнейшее развитие системы и ее использование также в качестве информационной базы при подготовке годовых отчетов академии о результатах учебной и научной деятельности.

В.А. Богатырев, доктор технических наук, профессор
С.В. Богатырев аспирант

КРИТЕРИИ ОПТИМАЛЬНОСТИ ОТКАЗОУСТОЙЧИВЫХ СЕРВЕРНЫХ СИСТЕМ

Введение

Современные корпоративные информационные системы и центры обработки данных строятся с использованием компьютерных сетевых технологий и включают в свой состав резервированные серверные системы различного функционального назначения (Web-серверы, почтовые серверы, серверы баз данных, FTP-серверы и т. п.), объединяемых в кластеры. При объединении серверных компьютерных систем в кластеры оптимизация включает определение кратности резервирования разнотипных серверов, при котором эффективность системы максимальна, а затрат на ее реализацию не превышают некоторый предел. Результат оптимизации во многом зависит от выбора критерия эффективности, формируемого с учетом требований по надежности, производительности (времени ожидания) и стоимости системы. В связи с этим рассматриваются варианты формирования частных и обобщенных критериев эффективности компьютерных систем, которые для отказоустойчивых систем должны характеризовать процесс деградации системы, вследствие накопления отказов.

Постановка задачи

Рассматривается компьютерная система, в которой за основу коммуникационной подсистемы взята древовидная топология. Для обеспечения

высокой надежности и производительности серверы резервируются и объединяются в группы - кластеры. Каждая группа серверов (кластер) подключается к сети через отдельный коммутатор (коммутационный узел) нижнего уровня (КНУ). Все КНУ взаимосвязаны через корневой коммутатор (коммутационный узел) верхнего уровня (КВУ). В системе выделяются N типов серверов, различающихся по функциональному назначению. Условие работоспособности системы заключается в работоспособности и доступности (связанности) хотя бы одного сервера каждого функционального типа, при не превышении среднего времени пребывания запросов заданный порог. Требуется оптимизировать распределение кратности резервирования узлов различных типов с учетом надежности производительности и стоимости реализации системы.

Структура исследуемой многоуровневой компьютерной системы с резервированием коммутационных узлов верхнего и нижнего уровней, а также серверов, объединяемых в кластеры, приведена на рис. 1.

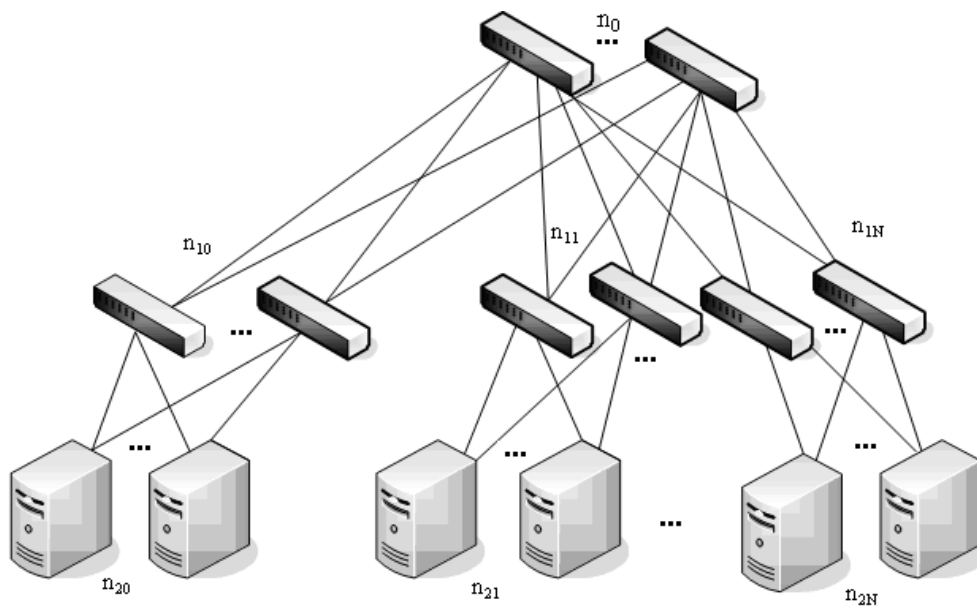


Рис. 1. Структура компьютерной системы

Для узлов каждого уровня будем считать заданными: v_{0i} и v_{1i} - среднее время передачи запроса к i -у серверу через КВУ и КНУ, v_{2i} - среднее время выполнения запроса в сервере i -й группы. Показатели надежности и стоимости реализации КВУ, КНУ и серверов соответствующих групп равны: p_0 , $(p_{10}, p_{11}, \dots, p_{1N})$, $(p_{20}, p_{21}, \dots, p_{2N})$, c_0 , $(c_{10}, c_{11}, \dots, c_{1N})$, $(c_{20}, c_{21}, \dots, c_{2N})$. Известны средняя интенсивность суммарного потока запросов λ и вероятности (доли) распределения запросов на обслуживание в каждую из N групп кластеров - $(b_1, b_2, b_3, \dots, b_N)$, $\sum_{i=1}^N b_i = 1$.

Требуется определить распределение числа узлов (кратностей резервирования) различных типов $n_0, n_{10}, n_{11}, \dots, n_{1N}, n_{20}, n_{21}, \dots, n_{2N}$, для которого достигается компромисс по максимуму надежности структуры и минимуму времени пребывания запросов, при ограничении средств C_0 на построение системы:

$P(n) \rightarrow \text{Max}; T_0(n) \rightarrow \text{Min}, T_1(n) \rightarrow \text{Min}, \dots, T_N(n) \rightarrow \text{Min}$ для $C(n) \leq C_0$.

При этом n_0 - кратность резервирования КВУ, а $(n_{10}, n_{11}, \dots, n_{1N})$ - кратности резервирования КНУ, используемых для подключения N групп серверов, кратность резервирования в каждой из которых $(n_{20}, n_{21}, \dots, n_{2N})$.

Решение задач векторной оптимизации

Известны подходы к решению задач векторной оптимизации, в том числе [1]:

- методы скалярной свертки, в том числе на основе:
- аддитивного критерия;
- мультипликативного критерия;
- определения отклонения от идеала;
- методы, основанные на ранжирование критериев, в том числе:
- метод главного критерия;
- лексикографическая оптимизация;
- метод последовательной уступки;
- методы, основанные на использовании функции полезности;
- минимаксные методы,

Перечисленные методы являются эвристическими и не исключают субъективизм лица принимающего (ЛПР). Субъективность в их основе может привести к выбору не лучшего решения.

Таким образом, целесообразен выбор объективного результирующего показателя эффективности, устанавливаемого на основе анализа назначения проектируемой системы, качество которой может характеризоваться единым показателем, зависимым от частных показателей. Сложность получения такого показателя сопряжена с тем, что требуется точно представлять назначение системы, как составной части надсистемы, и необходимы знания зависимости обобщенного показателя от частных. Как правило, результирующий показатель должен иметь некоторый физический смысл, подтверждающий соответствие системы ее функциональному назначению с позиции надсистемы [2].

В качестве объективного обобщенного показателя при решении векторной задачи оптимизации многоуровневой системы рассматривается возможность использования коэффициента сохранения эффективности. Коэффициент сохранения эффективности определяется как математиче-

ское ожидание нормированной эффективности (и среднего времени ожидания) для всех работоспособных состояний системы с учетом их вероятностей. Нормирование проводится относительно эффективности системы в исходном состоянии до возникновения отказов.

Оценка частных показателей

Возможности системы по обработке запросов определим по среднему времени пребывания запросов в системе. В простейшем случае каждый узел представим системой массового обслуживания типа М/М/1. Считая, что каждый запрос последовательно обслуживается, одним из исправных узлов на каждом уровне системы, среднее время пребывания запросов в системе оценим как:

$$T_i(\lambda, n) = \frac{v_{i0}}{1 - \sum_{j=0}^N \left(\frac{v_{j0} \lambda b_j}{n_0} \right)} + \frac{v_{1i}}{1 - \frac{v_{1i} \lambda b_i}{n_{1i}}} + \frac{v_{2i}}{1 - \frac{v_{2i} \lambda b_i}{n_{2i}}},$$

где $\sum_{j=0}^N \left(\frac{v_{j0} \lambda b_j}{n_0} \right) < 1$, $\frac{v_{1i} \lambda b_i}{n_{1i}} < 1$, $\frac{v_{2i} \lambda b_i}{n_{2i}} < 1$, $i=0, 1, \dots, N$ - условия стационарности.

Среднее время пребывания в системе запросов, адресуемых к разным группам серверов, вычисляется как:

$$\bar{T}(\lambda, n) = \sum_{i=0}^N b_i \left[\frac{v_{i0}}{1 - \sum_{j=0}^N \left(\frac{v_{j0} \lambda b_j}{n_0} \right)} + \frac{v_{1i}}{1 - \frac{v_{1i} \lambda b_i}{n_{1i}}} + \frac{v_{2i}}{1 - \frac{v_{2i} \lambda b_i}{n_{2i}}} \right].$$

Предполагая, что для функционирования системы достаточно исправности хотя бы одного узла каждого типа, вероятность работоспособного состояния системы вычисляется как:

$$P_c(n) = (1 - (1 - p_0)^{n_0}) \prod_{i=0}^N (1 - (1 - p_{1i})^{n_{1i}}) (1 - (1 - p_{2i})^{n_{2i}}).$$

При ограничении допустимого времени ожидания запросов, заданного для каждой группы серверов, вероятность работоспособности системы оценим как:

$$P_c(n) = \sum_{j=1}^{n_0} C_{n_0}^j p_0^j (1 - p_0)^{n_0 - j} \prod_{i=0}^L \left[\sum_{g=1}^{n_{1i}} \sum_{s=1}^{n_{2i}} \delta(i, j, g, s) C_{n_{1i}}^g p_{1i}^g (1 - p_{1i})^{n_{1i} - g} C_{n_{2i}}^s p_{2i}^s (1 - p_{2i})^{n_{2i} - s} \right],$$

$$\text{где } \delta(i, j, g, s) = \begin{cases} 0, & \text{if } T(i, j, g, s) > T_l(i), \\ 1, & \text{if } T(i, j, g, s) \leq T_l(i), \end{cases} \quad (1)$$

$T_i(i)$ - предельно допустимое среднее время пребывания в системе запросов к i -й группе серверов, $T(i, j, g, s)$ - среднее время пребывания в системе запросов к i -й группе серверов при исправности j КВУ, а в i -й группе работоспособности g КНУ и s серверов:

$$T(i, j, g, s) = \frac{v_{i0}}{1 - \sum_{i=0}^N \left(\frac{v_{i0} \lambda b_i}{j} \right)} + \frac{v_{1i}}{1 - \frac{v_{1i} \lambda b_i}{g}} + \frac{v_{2i}}{1 - \frac{v_{2i} \lambda b_i}{s}}. \quad (2)$$

Приведенные формулы учитывают возможность отказов коммуникационных узлов разного уровня системы, а линии связи рассматриваются как абсолютно надежные.

Если система работоспособна, когда запрос обслуживается за время, не превосходящее заданный предельный уровень (ограничение на допустимое время выполнения запросов), то надежность системы серверов (без учета коммуникационной подсистемы) вычисляется [3] как:

$$P(n_{20}, n_{21}, \dots, n_{2N}) = \sum_{k_1=1}^{n_{20}} \sum_{k_2=1}^{n_{21}} \sum_{k_3=1}^{n_{22}} \dots \sum_{k_N=1}^{n_{2N}} \delta(k_1, k_2, k_3, \dots, k_N) C_{n_{20}}^{k_1} C_{n_{21}}^{k_2} C_{n_{22}}^{k_3} \dots C_{n_{2N}}^{k_N} \times \\ \times p_{20}^{k_1} p_{21}^{k_2} p_{22}^{k_3} \dots p_{2N}^{k_N} (1 - p_{20})^{n_{20} - k_1} (1 - p_{21})^{n_{21} - k_2} (1 - p_{22})^{n_{22} - k_3} \dots (1 - p_{2N})^{n_{2N} - k_N},$$

где

$$\delta(k_1, k_2, k_3, \dots, k_N) = \begin{cases} 1, & \text{if } T(k_1, k_2, k_3, \dots, k_N) \leq T_0, \\ 0, & \text{if } T(k_1, k_2, k_3, \dots, k_N) > T_0, \end{cases} \quad (3)$$

$$T(k_1, k_2, k_3, \dots, k_N) = \sum_{i=0}^N b_i \left(\frac{v_{2i}}{1 - \frac{v_{2i} \lambda b_i}{k_i}} \right) \quad (4)$$

причем $T(k_1, k_2, k_3, \dots, k_N)$ - среднее время пребывания запросов в системе при исправности $k_1, k_2, k_3, \dots, k_N$ серверов соответствующих кластеров, T_0 - предельно допустимое среднее время пребывания запросов ко всем типам серверов.

Для системы содержащей три типа серверов ($N=3$) будем считать $p_{20} = 0,95, p_{21} = 0,9, p_{22} = 0,95, n_{20} = n_{22} = 4, n_{21} = 5, v_{20} = 1, v_{21} = 2, v_{22} = 1$ (с). Зависимость надежности $P(T_0)$ от предельно допустимого среднего времени пребывания запросов в системе T_0 приведена на рис 2 а, на котором кривые 1-3 соответствуют интенсивности входного потока $\lambda=1.3, 1.7, 1.85$ (1/с.). На рис 2 б кривыми 1-3 представлены потери надежности D из-за ограничений предельно допустимого времени ожидания, при $\lambda=1.3, 1.7, 1.85$ (1/с.).

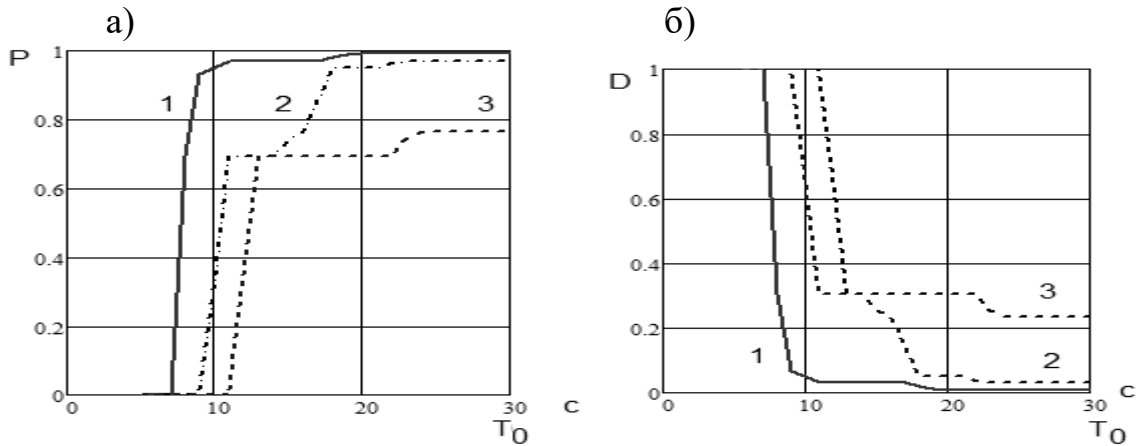


Рис. 2. Надежность серверной системы с учетом предельного времени пребывания запросов

Показатели сохранения эффективности

Степень сохранения эффективности каждого работоспособного состояния оценим величиной обратной отношению среднего времени пребывания запросов в этом и в исходном состоянии (когда отказов нет), тогда коэффициент сохранения эффективности всей системы (включающей систему серверов и коммуникационную подсистему) определим как математическое ожидание:

$$K(n) = \sum_{j=1}^{n_0} C_{n_0}^j p_0^j (1-p_0)^{n_0-j} \prod_{i=0}^L \left[\sum_{g=1}^{n_{i1}} \sum_{s=1}^{n_{i2}} \beta(i, j, g, s) C_{n_{i1}}^g p_{li}^g (1-p_{li})^{n_{i1}-g} C_{n_{i2}}^s p_{2j}^s (1-p_{2i})^{n_{i2}-s} \right],$$

где $\beta(i, j, g, s) = \delta(i, j, g, s) \frac{T_i(i)}{T(i, j, g, s)}$.

При этом $\delta(i, j, g, s)$ определяется по (1) а $T(i, j, g, s)$ по (2).

Если система работоспособна, когда запрос пребывает в ней время, не превосходящее заданный предельный уровень, то коэффициент сохранения эффективности серверной системы (без учета коммуникационной подсистемы) равен:

$$K(n_{20}, n_{21}, \dots, n_{2N}) = \sum_{k_1=1}^{n_{20}} \sum_{k_2=1}^{n_{21}} \sum_{k_3=1}^{n_{22}} \dots \sum_{k_N=1}^{n_{2N}} \beta(k_1, k_2, k_3, \dots, k_N) C_{n_{20}}^{k_1} C_{n_{21}}^{k_2} C_{n_{22}}^{k_3} \dots C_{n_{2N}}^{k_N} \times$$

$$\times p_{20}^{k_1} p_{21}^{k_2} p_{22}^{k_3} \dots p_{2N}^{k_N} (1-p_{20})^{n_{20}-k_1} (1-p_{21})^{n_{21}-k_2} (1-p_{22})^{n_{22}-k_3} \dots (1-p_{2N})^{n_{2N}-k_N},$$

где

$$\beta(k_1, k_2, k_3, \dots, k_N) = \delta(k_1, k_2, k_3, \dots, k_N) \frac{T(n_{20}, n_{21}, n_{22}, \dots, n_{2N})}{T(k_1, k_2, k_3, \dots, k_N)}.$$

Причем $\delta(k_1, k_2, k_3, \dots, k_N)$ и $T(k_1, k_2, k_3, \dots, k_N)$ вычисляются по (3), (4), а

$$T(n_{20}, n_{21}, \dots, n_{2N}) = \sum_{i=0}^N b_i \left(\frac{v_{2i}}{1 - \frac{v_{2i} \lambda b_i}{n_{2i}}} \right)$$

Зависимость коэффициента сохранения эффективности $K(T_0)$ от предельно допустимого времени пребывания запросов в системе T_0 приведена на рис 3, при $N=3$, $p_{20}=0,95, p_{21}=0,9, p_{22}=0,95, n_{20}=n_{21}=4, n_{22}=5, v_{20}=1, v_{21}=2, v_{22}=1$ (с). Кривые 1-3 соответствуют интенсивностям входного потока $\lambda=1,5, 1,7, 1,99$ (1/с) соответственно.

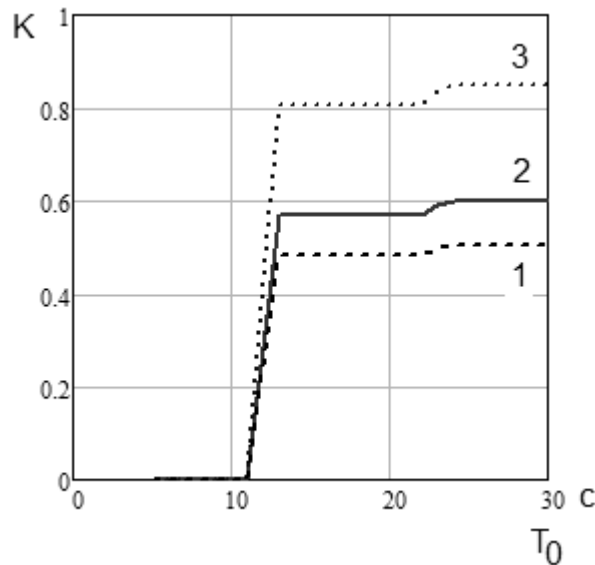


Рис. 3. Зависимость коэффициента сохранения эффективности $K(T_0)$ серверной подсистемы от предельно допустимого времени пребывания запросов в системе T_0 .

Заключение

Рассмотрены варианты формирования частных и обобщенных критериев при решении задачи векторной оптимизации серверных систем кластерной архитектуры с учетом их стоимости, надежности и времени обработки запросов. Показана целесообразность применения коэффициента сохранения эффективности в качестве объективного обобщенного критерия, при решении задач векторной оптимизации отказоустойчивых компьютерных систем.

Результаты исследований могут использоваться при проектировании и обосновании выбора структуры высоконадежных серверных систем, в том числе кластерной архитектуры.

Библиографический список

1. Черноруцкий И.Г. Методы принятия решений. СПб.: БХВ, 2005- 408 с.
2. Гуткин Л.С. Оптимизация радиоэлектронных устройств по совокупности показателей качества. М.: Сов. Радио, 1975.- 368 с.
3. Богатырев В.А. Богатырев С.В. Векторная оптимизация структуры кластера //Сборник научных трудов \\\\"Информационные системы и технологии: теория и практика\\ \\. : ЛТА, 2008. 19-27- с.

В.А. Богатырев, доктор технических наук, профессор
А.В.Осипов, студент

ЦЕЛОЧИСЛЕННАЯ ОПТИМИЗАЦИЯ МНОГОУРОВНЕВЫХ КОМПЬЮТЕРНЫХ СИСТЕМ В СРЕДЕ MATHCAD

Введение

Проектирование отказоустойчивых компьютерных систем сопряжено с решением задачи оптимального резервирования. При многоуровневой структуре компьютерных систем задача оптимизации сводится к определению кратности резервирования узлов каждого уровня, при котором эффективность системы максимальна, а затраты на ее реализацию не превышают допустимый предел.

Типовые варианты структуры компьютерных систем с резервированной коммуникационной подсистемой

Рассмотрим типовые структуры компьютерных систем с резервированием коммуникационной подсистемы, построенной на основе магистралей или коммутаторов [1].

Структура компьютерной системы, содержащей m равноправных компьютерных узлов, объединенных через n магистралей, приведена на рис 1 а), а через n коммутаторов - на рис 1 б). Подключение одного компьютерного узла к магистрали или порту коммутатора требует одного сетевого адаптера (СА). Будем считать, что компьютерная система работоспособна, если между любой парой из m компьютерных узлов осуществима взаимосвязь хотя бы по одной из n магистралей для систем по рис 1 а) или хотя бы через один коммутатор для систем по рис.1 б).

Для систем по рис 2, условие работоспособности коммуникационной подсистемы заключается в обеспечении взаимосвязи между компьютерными узлами разных уровней (например, между рабочими станциями и серверами, между контроллерами нижнего уровня управления и между компьютерами верхнего уровня управления), причем взаимосвязь между компьютерными узлами одного уровня может быть не обязательной.

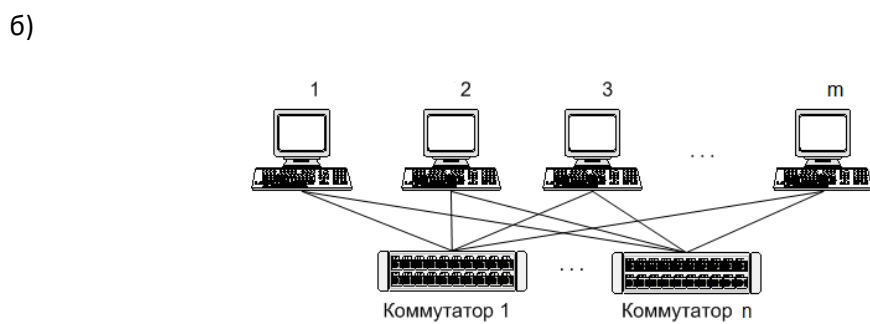
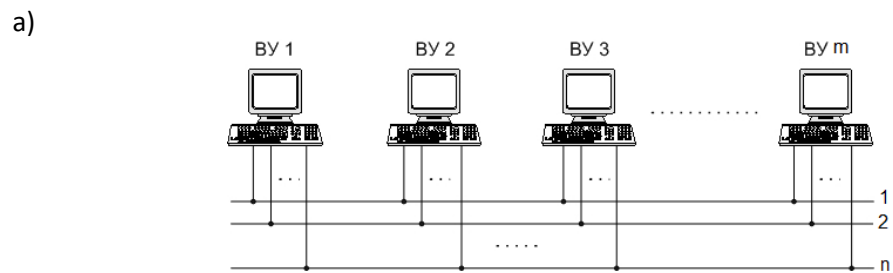


Рис. 1. Одноуровневая организация компьютерной системы

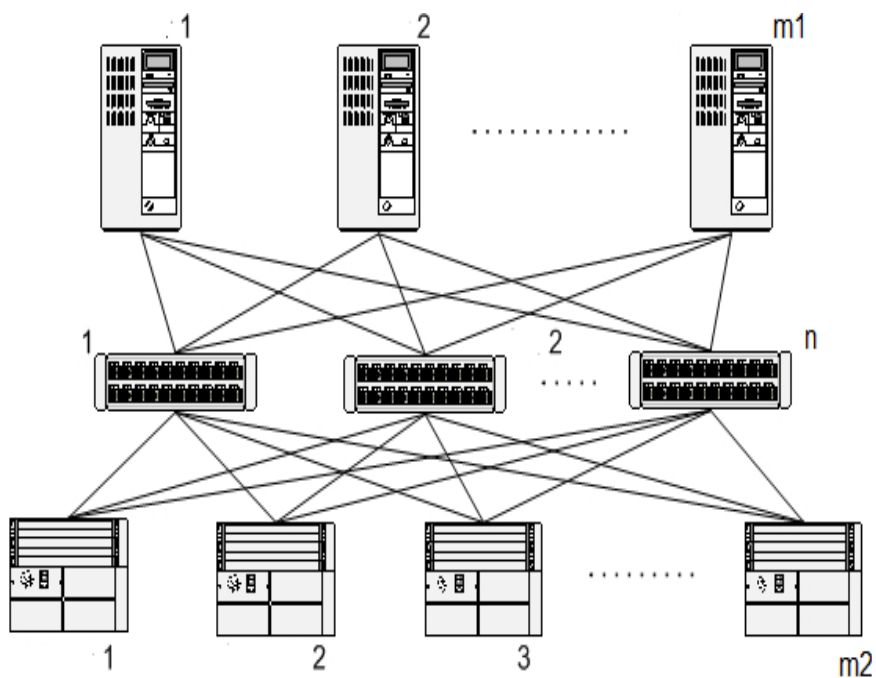


Рис.2. Двухуровневая организация компьютерной системы

Постановка задачи оптимизации

В качестве объекта оптимизации рассмотрим вычислительную систему, в которой выделены M уровней. Требуется найти число (кратность резервирования) узлов на каждом уровне (m_1, m_2, \dots, m_M) , при котором обеспечивается максимум надежности системы, минимум среднего времени ожидания, а стоимость реализации системы не должна превышать выделенных средств:

$$P(m_1, m_2, \dots, m_M) \rightarrow \max \text{ при } T(m_1, m_2, \dots, m_M) \rightarrow \min ,$$
$$C(m_1, m_2, \dots, m_M) = \sum_{i=1}^M m_i \leq C_0 .$$

Возможна также другая постановка задачи. Требуется найти число узлов на каждом уровне, при котором обеспечивается минимум стоимости реализации:

$$C(m_1, m_2, \dots, m_M) \rightarrow \min , \text{ при } P(m_1, m_2, \dots, m_M) \geq P_0 , T(m_1, m_2, \dots, m_M) \leq T_0 .$$

При этом T_0, P_0, C_0 – предельно допустимые значения среднего времени ожидания, надежности и стоимости системы.

При оптимизации будем считать заданными интенсивность входного потока запросов λ , показатели надежности узлов $p_1, p_2, p_3, \dots, p_M$ и средние времена выполнения запросов узлами каждого уровня $v_1, v_2, v_3, \dots, v_M$.

Особенность решаемой задачи является целочисленность.

При оптимизации системы по критерию «надежность» в качестве целевой функции может использоваться вероятность безотказной работы системы, которая для типовой многоуровневой системы представима в виде:

$$P(n_1, n_2, n_3) = \left(\sum_{i=1}^{n_1} C_{n_1}^i P_1^i (1 - P_1)^{n_1-i} \right)^m \left(\sum_{i=1}^{n_2} C_{n_2}^i P_2^i (1 - P_2)^{n_2-i} \right) \left(\sum_{i=1}^{n_3} C_{n_3}^i P_3^i (1 - P_3)^{n_3-i} \right), \quad (1)$$

где: P_1, P_2, P_3 – вероятность безотказной работы контроллера, коммутатора и управляющего компьютера, а n_1, n_2, n_3 их число.

Решение задачи целочисленной оптимизации

Пакет MathCAD обладает встроенными средствами решения задач оптимизации. Для этих целей в MathCAD имеются функции: $\text{Minimize}(f, x_1, x_2, \dots)$, $\text{Maximize}(f, x_1, x_2, \dots)$, которые возвращают значения переменных x_1, x_2, \dots , удовлетворяющие условиям в блоке Given и обеспечивающие минимальное или максимальное значение функция f .

Однако, опыт работы в системе MathCAD, показал, что встроенные средства MathCAD не позволяют найти целочисленное решение оптимизационных задач и найти решение, если аргумент x оптимизируемой функции находится над знаком суммы $(\sum_{i=i_0}^x y)$.

Для преодоления указанных ограничений предлагается целочисленного решения поставленной оптимизационной задачи средствами программирования системы MathCAD.

Ниже приведен фрагмент программы поиска оптимального числа узлов компьютерной системы, обеспечивающего максимум надежности системы с учетом ограничений средств на ее реализацию:

$$fc(x) = \begin{cases} (c_1 m) & \text{if } x=1 \\ c_2 & \text{if } x=2 \\ c_3 & \text{if } x=3 \end{cases}$$

$$\begin{aligned} \text{rank1} &\leftarrow 1 \text{ if } x = (fc(1) > fc(2)) \wedge (fc(1) > fc(3)) \\ \text{rank1} &\leftarrow 2 \text{ if } x = (fc(2) > fc(1)) \wedge (fc(2) > fc(3)) \\ \text{rank1} &\leftarrow 3 \text{ if } x = (fc(3) > fc(2)) \wedge (fc(3) > fc(1)) \\ \text{rank2} &\leftarrow 1 \text{ if } x = [(fc(1) < fc(2)) \wedge (fc(1) > fc(3))] \vee [(fc(1) < fc(3)) \wedge (fc(1) > fc(2))] \\ \text{rank2} &\leftarrow 2 \text{ if } x = [(fc(2) < fc(1)) \wedge (fc(2) > fc(3))] \vee [(fc(2) < fc(3)) \wedge (fc(2) > fc(1))] \\ \text{rank2} &\leftarrow 3 \text{ if } x = [(fc(3) < fc(2)) \wedge (fc(3) > fc(1))] \vee [(fc(3) < fc(1)) \wedge (fc(3) > fc(2))] \\ \text{rank3} &\leftarrow 1 \text{ if } x = (fc(1) < fc(2)) \wedge (fc(1) < fc(3)) \\ \text{rank3} &\leftarrow 2 \text{ if } x = (fc(2) < fc(1)) \wedge (fc(2) < fc(3)) \\ \text{rank3} &\leftarrow 3 \text{ if } x = (fc(3) < fc(2)) \wedge (fc(3) < fc(1)) \end{aligned}$$

$$\begin{aligned} P_0 &\leftarrow 0 \\ \text{for } i_{\text{rank1}} &\in g(\text{rank1}), g(\text{rank1})+1 \cdot \left(\frac{c_0}{fc(\text{rank1})} \right) \\ \text{for } i_{\text{rank2}} &\in g(\text{rank2}), g(\text{rank2}) \cdot \left(\frac{c_0 - fc(\text{rank1})i_{\text{rank1}}}{fc(\text{rank2})} \right) \text{ if } \left(\frac{c_0 - fc(\text{rank1})i_{\text{rank1}}}{fc(\text{rank2})} \right) \geq 1 \\ \text{for } i_{\text{rank3}} &\in g(\text{rank3}), g(\text{rank3}) \cdot \left(\frac{c_0 - fc(\text{rank1})i_{\text{rank1}} - fc(\text{rank2})i_{\text{rank2}}}{fc(\text{rank3})} \right) \text{ if } \left(\frac{c_0 - fc(\text{rank1})i_{\text{rank1}} - fc(\text{rank2})i_{\text{rank2}}}{fc(\text{rank3})} \right) \geq 1 \\ y &= \begin{cases} i_1 \leftarrow 1 \text{ if } \text{rank1} = 1 \\ i_1 \leftarrow 2 \text{ if } \text{rank2} = 1 \\ i_1 \leftarrow 3 \text{ if } \text{rank3} = 1 \\ i_2 \leftarrow 1 \text{ if } \text{rank1} = 2 \\ i_2 \leftarrow 2 \text{ if } \text{rank2} = 2 \\ i_2 \leftarrow 3 \text{ if } \text{rank3} = 2 \\ i_3 \leftarrow 1 \text{ if } \text{rank1} = 3 \\ i_3 \leftarrow 2 \text{ if } \text{rank2} = 3 \\ i_3 \leftarrow 3 \text{ if } \text{rank3} = 3 \end{cases} \\ \text{if } P_0 &< P(i_1, i_2, i_3) \\ &\begin{cases} y_0 \leftarrow i_1 \\ y_1 \leftarrow i_2 \\ y_2 \leftarrow i_3 \\ P_0 \leftarrow P(i_1, i_2, i_3) \end{cases} \end{aligned}$$

Результаты решения задачи сведены в таблицу.

Т а б л и ц а

Результаты расчетов

Исходные условия								Результат расчетов			
Кол-во ОУ	Стоимость элементов системы			Вероятность безотказной работы элементов			Лимит стоимости системы c_0	Кратность элементов			Вероятность безотказной работы P
	c_1	c_2	c_3	P_1	P_2	P_3		n_1	n_2	n_3	
2	400	500	600	0.85	0.75	0.80	7000	3	4	4	0.988
5	200	350	750	0.82	0.78	0.90	10000	5	5	4	0.998
10	200	350	250	0.80	0.79	0.92	15000	6	5	5	0.999

Результаты исследования представлены графиками на рис 3-4, на которых кривые 1-3 соответствуют числу контроллеров, коммутаторов и компьютеров. Рис. 3 показывает, как изменяется число узлов каждого уровня, соответствующих максимуму целевой функции, в зависимости от вероятности безотказной работы контроллеров.

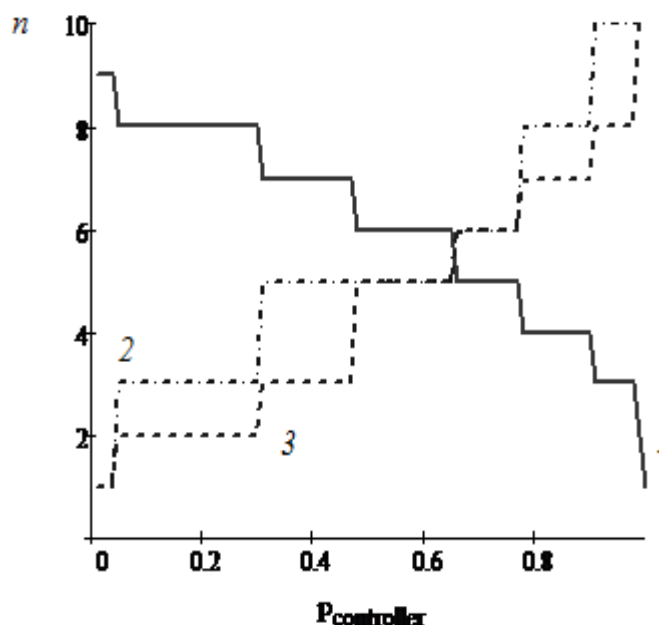


Рис. 3. Зависимость оптимального числа элементов системы от вероятности безотказной работы контроллеров

На рис. 4 отражена зависимость оптимального количества контроллеров, коммутаторов и управляющих компьютеров от стоимости коммута-

торов. Рис. 4 показывает, что при повышении стоимости коммутаторов их количество в оптимальной конфигурации системы уменьшается.

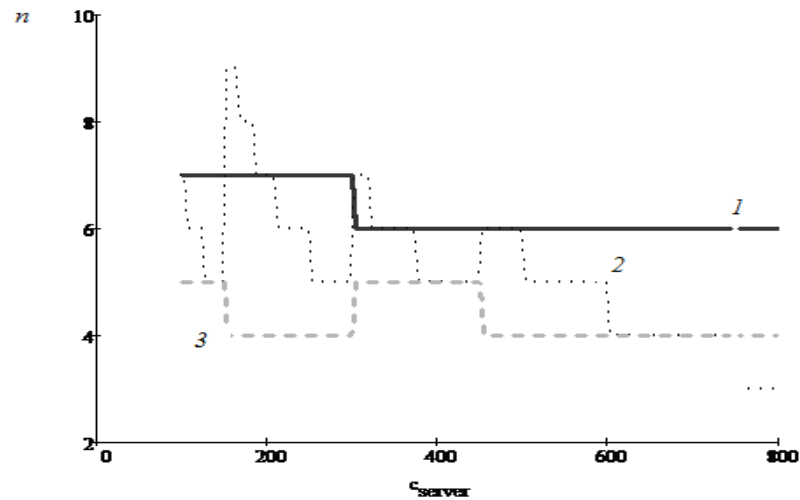


Рис. 4. Зависимость оптимального числа элементов системы от стоимости серверов

Рис. 5 отражает зависимость времени выполнения алгоритма (в секундах) от общего количества элементов в системе .

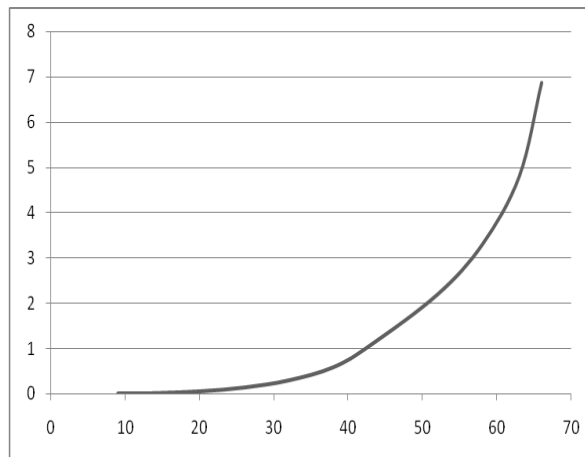


Рис. 5. Зависимость времени выполнения алгоритма от общего количества элементов в системе

Заключение

Таким образом, предложен алгоритм целочисленного решения задачи оптимального проектирования вычислительных систем. В отличие от использования традиционных для оптимизации в среде MathCAD функций $\text{Minimize}(f, x_1, x_2, \dots)$ / $\text{Maximize}(f, x_1, x_2, \dots)$ предлагаемый алгоритм позволяет найти целочисленное решение, если искомые переменные находятся над знаком суммы.

Библиографический список

1. Швецов Д. SCADA-система GENESIS32 в сквозной автоматизации производства // Рациональное управление предприятием \ 2007 № 6, с 15-19.
2. Богатырев В.А., Богатырев С.В. Векторная оптимизация структуры кластера // Сборник научных трудов \ Информационные системы и технологии: теория и практика \ : ЛТА, 2008. 19-27- с.
3. Богатырев В.А. Оценка надежности и оптимальное резервирование кластерных компьютерных систем // Приборы и системы. Управление, контроль, диагностика \ 2006 №10 с 18-21
4. Макаров Е. Г. Mathcad: учебный курс – СПб.: Питер, 2009 г, 384 с.

С.В.Гуров, доктор технических наук, профессор

ОЦЕНКА НАДЕЖНОСТИ СЛОЖНОЙ СИСТЕМЫ С ЗАВИСИМЫМИ ЭЛЕМЕНТАМИ

Показатели надежности технической системы обычно определяются в предположении независимости входящих в нее элементов. Это значит, что отказ какого-либо элемента никак не влияет на надежность других элементов. Данное предположение лежит в основе математического аппарата теории надежности.

На самом деле, в результате отказов некоторых элементов неизбежно возникает дополнительная нагрузка на элементы, находящиеся в работоспособном состоянии. Как следствие, происходит изменение надежности элементов. Указанному явлению, называемому последствием отказов, в той или иной степени подвержена любая резервированная система. Интенсивности отказов элементов при этом изменяются в зависимости от времени эксплуатации системы, и сами элементы системы оказываются зависимыми друг от друга. В результате эффекта последствия происходит более интенсивное старение системы, и сокращение времени ее «жизни».

Изменение нагрузки на элементы системы и их надежности может быть связано также с условиями эксплуатации системы и влиянием ряда внешних факторов. При этом возможны ситуации, как увеличения, так и уменьшения нагрузки (нагрузочное резервирование). Может иметь место и колебательный характер изменения нагрузки. По признаку зависимости от времени нагрузка может быть дискретной, непрерывной или смешанной.

Методов расчета надежности систем с зависимыми элементами в настоящее время практически не существует. Исключением являются лишь системы с последствием отказов, когда рассматривается только дублирование или троирование одинаковых по надежности элементов с

экспоненциальными распределениями времени до отказа. При этом остается открытым вопрос, по каким законам происходит изменение интенсивности отказов элементов.

Общая ситуация в научной литературе практически не изучалась. Это связано с тем, что не понятно, каким образом возникающая нагрузка на элементы системы оказывает влияние на законы распределения соответствующих случайных величин. Последние работы в этом направлении, в какой-то степени, сняли указанные препятствия и позволили исследовать надежность систем, как с увеличением, так и с уменьшением нагрузки на их элементы. Надо сказать, что расчет надежности систем с последствием отказов в общем случае невозможно выполнить без применения надежных программных средств.

Математическое описание работы системы с зависимыми элементами

Пусть $E = E_+ \cup E_-$ – множество всех состояний системы, причем E_+ – множество всех исправных состояний, а E_- – множество всех состояний отказа.

Тогда надежность системы по критерию вероятности безотказной работы можно рассчитать по формуле

$$P_c(t) = p_0(t) + \sum_{(j_1, j_2, \dots, j_i) \in E_+} p_{j_1, j_2, \dots, j_i}(t), \quad (1)$$

где $p_0(t)$ – вероятность того, что в момент времени t все элементы системы работоспособны,

$p_{j_1, j_2, \dots, j_i}(t)$ – вероятность события, состоящего в том, что до момента времени t отказало ровно i элементов с номерами j_1, j_2, \dots, j_i в указанной здесь последовательности. Суммирование в (1) производится по всем исправным состояниям вида (j_1, j_2, \dots, j_i) за исключением начального состояния, когда все элементы системы исправны.

Выпишем соотношения для указанных вероятностей событий. Вероятность того, что все элементы системы работоспособны в момент времени t , очевидно, равна

$$p_0(t) = \prod_j \bar{F}_j(t), \quad (2)$$

где $\bar{F}_j(t)$ – вероятность безотказной работы j -го элемента, и произведение в (2) вычисляется по всем элементам системы.

Вероятность последовательного отказа элементов с номерами j_1, j_2, \dots, j_i до момента времени t найдем по формуле полной вероятности:

$$p_{j_1, j_2, \dots, j_i}(t) = \int_{t_1 \leq t_2 \leq \dots \leq t_i \leq t} f_{j_1}(t_1) f_{j_2}(t_2, t_1) f_{j_3}(t_3, t_1, t_2) \dots f_{j_i}(t_i, t_1, t_2, \dots, t_{i-1}) \prod_{j \neq j_1, j_2, \dots, j_i} \bar{F}_j(t, t_1, \dots, t_i) dt_1 \dots dt_i \quad (3)$$

Здесь и далее используются следующие обозначения:

t_i – момент отказа элемента с номером j_i . В этот момент происходит изменение интенсивностей отказов других элементов;

$f_{j_1}(t)$ – плотность распределения времени безотказной работы элемента с номером j_1 ;

$f_{j_2}(t, t_1)$ – плотность распределения времени безотказной работы элемента с номером j_2 , если в момент t_1 произошел отказ элемента с номером j_1 . В момент t_1 на элемент с номером j_2 изменилась нагрузка.

$f_{j_3}(t, t_1, t_2)$ – плотность распределения времени безотказной работы элемента с номером j_3 , если в момент t_1 произошел отказ элемента с номером j_1 , а в момент t_2 – отказ элемента с номером j_2 . При этом в указанные моменты времени дважды изменилась нагрузка на элемент с номером j_3 , и т.д.;

$f_{j_i}(t, t_1, t_2, \dots, t_{i-1})$ – плотность распределения времени безотказной работы элемента с номером j_i , если в моменты t_1, t_2, \dots, t_{i-1} произошли отказы элементов с номерами j_1, j_2, \dots, j_{i-1} соответственно. В указанные моменты времени менялась нагрузка на элемент с номером j_i .

Аналогичный смысл имеет и функция $\bar{F}_j(t, t_1, t_2, \dots, t_i)$. Это вероятность безотказной работы элемента с номером j , если в моменты t_1, t_2, \dots, t_i произошли отказы элементов с номерами j_1, j_2, \dots, j_i соответственно. Как и плотности распределения, эти вероятности также зависят от предыстории работы системы.

Заметим, что формулы (1)-(3) справедливы для любой дополнительной нагрузки на исправные элементы, вызванной отказом некоторых элементов системы. Если условия функционирования системы с течением времени ухудшаются, то коэффициенты нагрузки больше единицы и надежность системы убывает. Если условия функционирования системы со временем облегчаются, то надежность системы должна возрасти, при этом коэффициенты нагрузки k_i меньше единицы. Это может происходить, например, при уменьшении интенсивности отказов элементов, когда имеет место нагрузочное резервирование. Указанный вид резервирования заранее предусматривает возможность работы системы при повышенных нагрузках. Физически нагрузочное резервирование реализуется путем облегчения режимов работы элементов [1]. Однако, как и структурное резервирование, оно неизбежно сопровождается увеличением объемов оборудования.

Предположим, что при отсутствии отказов нагрузка на элементы системы не меняется и равна $k_0 = 1$.

После отказа любого одного элемента нагрузка на остальные элементы изменяется и становится равной k_1 , после отказа любых двух элементов нагрузка на остальные элементы становится равной k_2 , и т.д., после отказа i элементов нагрузка на исправные элементы становится равной k_i .

Метод расчета приведенных здесь условных плотностей и вероятностей основан на результатах работы, в которой анализируется надежность систем с кусочно-постоянной нагрузкой. Дадим краткое описание этого метода в предположении, что увеличение нагрузки на элемент в k раз вызывает увеличение интенсивности отказа этого элемента также в k раз.

Аргумент t_i искомой функции $\bar{F}_j(t, t_1, t_2, \dots, t_i)$ есть момент последнего отказа, после которого j -ый элемент меняет свою нагрузку со значения k_{i-1} на k_i . Поскольку $t_i \leq t$, то условная вероятность вычисляется через безусловную на основании равенства

$$\bar{F}_j(t, t_1, t_2, \dots, t_i) = \bar{F}_j^{k_i}(t - x_i), \quad (4)$$

где параметр x_i определяется рекуррентно через x_1, x_2, \dots, x_{i-1} , исходя из следующих уравнений

$$\begin{aligned} \bar{F}_j^{k_1}(t_1 - x_1) &= \bar{F}_j(t_1), \\ \bar{F}_j^{k_2}(t_2 - x_2) &= \bar{F}_j^{k_1}(t_2 - x_1), \\ &\dots \\ \bar{F}_j^{k_i}(t_i - x_i) &= \bar{F}_j^{k_{i-1}}(t_i - x_{i-1}). \end{aligned}$$

Дифференцируя вероятность $\bar{F}_j(t, t_1, t_2, \dots, t_i)$ по t , найдем выражение условной плотности распределения через безусловную плотность и безусловную вероятность безотказной работы j -го элемента:

$$f_j(t, t_1, t_2, \dots, t_i) = k_i \bar{F}_j^{k_i-1}(t - x_i) f_j(t - x_i). \quad (5)$$

Таким образом, соотношения (1)-(3) позволяют рассчитать вероятность безотказной работы сложной системы с произвольными распределениями времени работы до отказов ее элементов при весьма общих предположениях об их зависимости. Основные математические трудности расчета надежности связаны с необходимостью решения алгебраических уравнений и вычисления кратных интегралов. Первая трудность легко преодолевается, если в библиотеке вспомогательных программ предусмотреть расчет законов распределения вероятностей требуемых распределений и расчет обратных функций для вероятностей безотказной работы. Для вычисления кратных интегралов удобно использовать метод статистических испытаний, состоящий в разыгрывании на ЭВМ случайных точек (x_1, x_2, \dots, x_i) равномерно распределенных на многомерном симплексе $x_1 + x_2 + \dots + x_i \leq t$. Этот вопрос рассмотрен в работе [1]. Важный вопрос состоит в количестве проводимых испытаний. Он решается следующим образом. Число испыта-

ний N принимается таким, чтобы была гарантирована требуемая точность расчетов вероятности безотказной работы $P(t)$ системы без последствия двумя способами:

- по формуле (1),
- методом статистических испытаний для случая отсутствия дополнительной нагрузки на элементы системы при отказе других элементов, то есть когда $k(t) \equiv 1$.

Вероятности состояний системы из элементов с экспоненциальными распределениями

Если законы распределения времени до отказа элементов экспоненциальные, то можно получить аналитические выражения для характеристик надежности системы, не требующие сложных математических расчетов. Предположим, что вначале функционирования все элементы системы имеют интенсивность отказов, равную λ . Тогда из (4) следует, что

$$\begin{aligned} k_1 x_1 &= (k_1 - 1)t_1, \\ k_2 x_2 &= (k_2 - k_1)t_2 + k_1 x_1, \\ &\dots \\ k_i x_i &= (k_i - k_{i-1})t_i + k_{i-1} x_{i-1}. \end{aligned}$$

Суммируя эти соотношения, получим

$$k_i x_i = \sum_{j=1}^i (k_j - k_{j-1})t_j$$

Тогда условная вероятность представляется в виде

$$\bar{F}(t, t_1, t_2, \dots, t_i) = e^{-k_i \lambda (t - x_i)} = e^{-\lambda \left(k_i t - \sum_{j=1}^i (k_j - k_{j-1}) t_j \right)},$$

а условная плотность распределения – в виде

$$f(t, t_1, t_2, \dots, t_i) = \lambda k_i e^{-\lambda \left(k_i t - \sum_{j=1}^i (k_j - k_{j-1}) t_j \right)}.$$

Вероятность отказа элементов с номерами j_1, j_2, \dots, j_i до момента времени t в указанном здесь порядке равна

$$P_{j_1, j_2, \dots, j_i}(t) = \lambda^i k_1 \dots k_{i-1} \int \dots \int_{t_1 \leq t_2 \leq \dots \leq t_i \leq t} e^{-\lambda S(t, t_1, t_2, \dots, t_i)} dt_1 \dots dt_i,$$

где

$$S(t, t_1, t_2, \dots, t_i) = (n - i)k_i t - (n - i) \sum_{j=1}^i (k_j - k_{j-1}) t_j + \sum_{m=1}^i \left(k_{m-1} t_m - \sum_{j=1}^{m-1} (k_j - k_{j-1}) t_j \right).$$

Преобразуя последнее выражение, получим

$$S(t, t_1, t_2, \dots, t_i) = (n - i)k_i t + \sum_{j=1}^i \left((n - j + 1)k_{j-1} - (n - j)k_j \right) t_j.$$

Следовательно,

$$p_{j_1, j_2, \dots, j_i}(t) = \lambda^i k_1 \dots k_{i-1} e^{-\lambda c_0 t} \int \dots \int_{t_1 \leq t_2 \leq \dots \leq t_i \leq t} e^{-\lambda \sum_{j=1}^i c_j t_j} dt_1 \dots dt_i, \quad (6)$$

$$\text{где } c_0 = (n-i)k_i, \quad c_j = (n-j+1)k_{j-1} - (n-j)k_j, \quad j = 1, 2, \dots, i. \quad (7)$$

$$p_{j_1, j_2, \dots, j_i}\left(\frac{t}{\lambda}\right) = k_1 \dots k_{i-1} e^{-c_0 t} \int \dots \int_{t_1 \leq t_2 \leq \dots \leq t_i \leq t} e^{-\sum_{j=1}^i c_j t_j} dt_1 \dots dt_i. \quad (8)$$

Таким образом, вычисление вероятности свелось к вычислению кратного интеграла от функции специального вида – экспоненты линейной функции.

Вычисление кратных интегралов для экспоненциальных распределений

Получим алгоритм вычисления кратного интеграла вида

$$p_i(t) = \int \dots \int_{t_1 \leq t_2 \leq \dots \leq t_i \leq t} e^{-(c_1 t_1 + c_2 t_2 + \dots + c_i t_i)} dt_1 \dots dt_i.$$

Поскольку

$$p_i(t) = \int_0^t dt_i \int_0^{t_i} dt_{i-1} \dots \int_0^{t_2} dt_1 e^{-(c_1 t_1 + c_2 t_2 + \dots + c_i t_i)},$$

то имеет место рекуррентное соотношение

$$p_i(t) = \int_0^t e^{-c_i t_i} p_{i-1}(t_i) dt_i. \quad (9)$$

В преобразовании Лапласа получим

$$\hat{p}_i(z) = \frac{1}{z} \hat{p}_{i-1}(z + c_i).$$

Применим аналогичное соотношение к функции \hat{p}_{i-1} , тогда

$$\hat{p}_i(z) = \frac{1}{z(z + c_i)} \hat{p}_{i-2}(z + c_i + c_{i-1}).$$

Последовательно повторяя эту процедуру, получим

$$\hat{p}_i(z) = \frac{1}{z(z + c_i)(z + c_i + c_{i-1}) \dots (z + c_i + c_{i-1} + \dots + c_1)}. \quad (10)$$

Чтобы найти оригинал по дробно-рациональному изображению (10) необходимо представить выражение (10) в виде суммы простейших дробей. Этот вопрос теоретически хорошо изучен. Однако, в случае кратных корней знаменателя найти коэффициенты разложения оказывается непросто вычислительной задачей. Пусть знаменатель дроби имеет k различных корней z_1, z_2, \dots, z_k с кратностями r_1, r_2, \dots, r_k соответственно, и пусть

$$\hat{p}(z) = \prod_{i=1}^k \frac{1}{(z - z_i)^{r_i}}. \quad (11)$$

Представим (10) в виде суммы простейших дробей:

$$\hat{p}(z) = \sum_{i=1}^k \sum_{j=0}^{r_i-1} \frac{a_{ij}}{(z - z_i)^{r_i-j}} \quad (12)$$

с некоторыми неопределенными коэффициентами a_{ij} . Если все корни знаменателя простые, то есть $r_1 = r_2 = \dots = r_k = 1$, то

$$\hat{p}(z) = \sum_{i=1}^k \frac{a_{i0}}{z - z_i},$$

и коэффициенты разложения определяются формулами

$$a_{i0} = \prod_{\substack{l=1 \\ l \neq i}}^k \frac{1}{z_i - z_l}, \quad i = 1, 2, \dots, k.$$

Пусть теперь корни знаменателя – кратные. Тогда коэффициенты разложения a_{ij} можно определить по формулам

$$a_{ij} = \frac{1}{j!} \lim_{z \rightarrow z_i} \frac{d^j}{dz^j} \left(\prod_{\substack{l=1 \\ l \neq i}}^k \frac{1}{(z - z_l)^{r_l}} \right), \quad i = 1, 2, \dots, k, \quad j = 0, 1, 2, \dots, r_i - 1.$$

Отсюда коэффициенты a_{ij} можно найти аналитически, однако, они представляются многомерными суммами, что сильно увеличивает время расчетов. Можно пойти по другому пути и свести задачу нахождения коэффициентов к решению системы линейных алгебраических уравнений. Рассмотрим этот путь более подробно.

Из представления дроби (11) в виде суммы (12) следует тождество

$$\sum_{i=1}^k \sum_{j=0}^{r_i-1} a_{ij} (z - z_i)^j \prod_{\substack{l=1 \\ l \neq i}}^k (z - z_l)^{r_l} = 1, \quad (13)$$

справедливое при всех z . Число неизвестных коэффициентов равно $n = \sum_{i=1}^k r_i$, поэтому для их определения надо составить систему из n линейных уравнений. Такую систему получим, подставив в равенстве (13) в качестве z n произвольных различных значений. Тогда решение построенной таким образом системы уравнений позволит определить искомые коэффициенты a_{ij} .

Применяя к (11) обратное преобразование Лапласа, получим

$$p(t) = \sum_{i=1}^k \sum_{j=0}^{r_i-1} a_{ij} \frac{t^{r_i-j-1}}{(r_i - j - 1)!} e^{z_i t}. \quad (14)$$

Рассмотренный метод определения оригинала по дробно-рациональному изображению оказывается достаточно эффективным при реализации его на компьютере. Разложение функции (11) в виде суммы простейших дробей (12) реализовано программой с исходным кодом `decomposition_crush.cpp`.

Наработка до отказа системы с зависимыми элементами

Поскольку средняя наработка на отказ равна

$$T_1 = \int_0^{\infty} P_c(t) dt,$$

то на основе формулы (1), позволяющей рассчитать вероятность безотказной работы системы с зависимыми элементами, получим

$$T_1 = \int_0^{\infty} p_0(t) dt + \frac{1}{\lambda} \sum_{(j_1, j_2, \dots, j_i) \in E_+} \int_0^{\infty} p_{j_1, j_2, \dots, j_i} \left(\frac{t}{\lambda} \right) dt.$$

Так как для экспоненциальных распределений $p_0(t) = e^{-n\lambda t}$, а

$$\int_0^{\infty} p_{j_1, j_2, \dots, j_i} \left(\frac{t}{\lambda} \right) dt = k_1 k_2 \dots k_{i-1} \hat{p}_i(c_0) = \frac{k_1 k_2 \dots k_{i-1}}{c_0(c_0 + c_i)(c_0 + c_i + c_{i-1}) \dots (c_0 + c_i + c_{i-1} + \dots + c_1)},$$

то наработка на отказ оказывается равной

$$T_1 = \frac{1}{n\lambda} + \frac{1}{\lambda} \sum_{(j_1, j_2, \dots, j_i) \in E_+} \frac{k_1 k_2 \dots k_{i-1}}{c_0(c_0 + c_i)(c_0 + c_i + c_{i-1}) \dots (c_0 + c_i + c_{i-1} + \dots + c_1)}.$$

Отсюда следует, что вычисление средней наработки требует лишь перебора исправных состояний системы, определением путей графа, ведущих из начального во все исправные состояния и суммирования слагаемых по этим путям.

Пример расчет вероятности состояния системы с зависимыми элементами

Дана система, структурная схема надежности которой изображена на рис.1. Элементы системы имеют одинаковую надежность с интенсивностью отказов λ . Требуется определить вероятность одного из исправных состояний этой системы для трех условий эксплуатации:

- при отказе элементов нагрузка и интенсивность отказов остальных элементов не изменяется, то есть $k_1 = k_2 = k_3 = k_4 = 1$;
- в результате отказа хотя бы одного элемента нагрузка на остальные элементы становится равной $k_1 = 2$, и после этого она не изменяется, то есть $k_2 = k_3 = k_4 = 2$;
- в результате отказа одного, двух, трех или четырех элементов интенсивность отказов остальных элементов возрастает соответственно в $k_1 = 2$, $k_2 = 2,4$, $k_3 = 3$, $k_4 = 4$ раза.

В системе $n = 7$ элементов. Рассмотрим следующий порядок отказов $i = 4$ элементов: 6,4,1,3. Вычислим вероятность $p_{6,4,1,3}(t)$ этого состояния в момент времени t . Расчеты содержатся в табл.1 и 2. Они проведены для трех заданных условий эксплуатации системы.

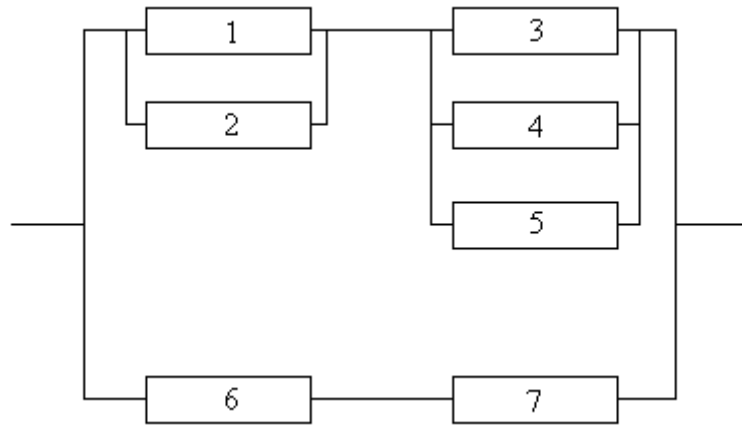


Рис.1

В соответствии с формулами (7) в табл.1 вычислены коэффициенты

$$\begin{aligned}
 c_0 &= (n-4)k_4, \\
 c_1 &= n - (n-1)k_1, \\
 c_2 &= (n-1)k_1 - (n-2)k_2, \\
 c_3 &= (n-2)k_2 - (n-3)k_3, \\
 c_4 &= (n-3)k_3 - (n-4)k_4.
 \end{aligned}$$

Согласно (8) вероятность искомого состояния равна

$$p_{6,4,1,3}(t) = k_1 k_2 k_3 e^{-c_0 t} p(\lambda t),$$

где функция $p(t)$ в преобразовании Лапласа на основе (9) имеет вид, содержащийся в табл.1.

Т а б л и ц а 1

Ус- ло- вие	c_0	c_1	c_2	c_3	c_4	$\hat{p}(z)$
1	3	1	1	1	1	$\frac{1}{z(z+1)(z+2)(z+3)(z+4)} = \frac{a_{10}}{z} + \frac{a_{20}}{z+1} + \frac{a_{30}}{z+2} + \frac{a_{40}}{z+3} + \frac{a_{50}}{z+4}$
2	6	-5	2	2	2	$\frac{1}{z(z+2)(z+4)(z+6)(z+1)} = \frac{a_{10}}{z} + \frac{a_{20}}{z+2} + \frac{a_{30}}{z+4} + \frac{a_{40}}{z+6} + \frac{a_{50}}{z+1}$
3	12	-5	0	0	0	$\frac{1}{z^4(z+5)} = \frac{a_{10}}{z^4} + \frac{a_{11}}{z^3} + \frac{a_{12}}{z^2} + \frac{a_{13}}{z} + \frac{a_{20}}{z+5}$

В табл.2 получены коэффициенты разложения рациональной дроби на простейшие. Они совпадают с коэффициентами, полученными по программе `decomposition_crush.cpp`. В этой же таблице получен оригинал $p(t)$ с использованием формулы (14) и вероятность $p_{6,4,1,3}(t)$ состояния 6,4,1,3.

Т а б л и ц а 2

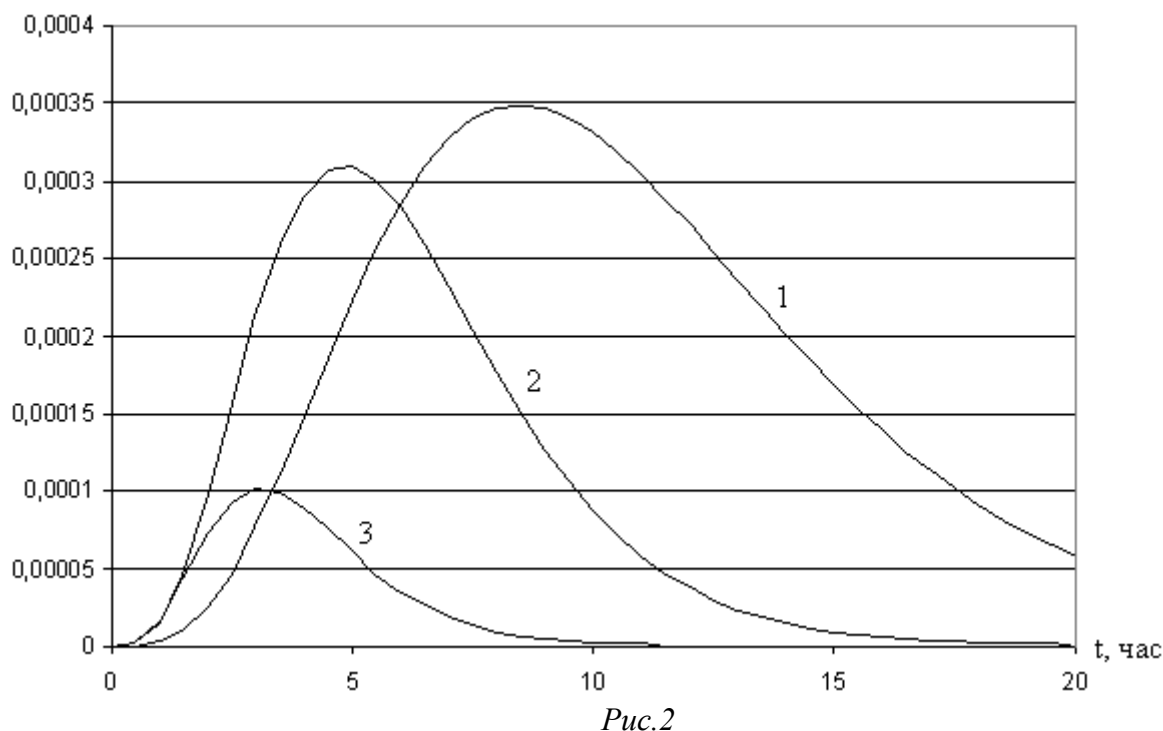
Усло- вие	Коэффициенты разложения, $p(t)$, $p_{6,4,1,3}(t) = k_1 k_2 k_3 e^{-c_0 \lambda t} p(\lambda t)$
1	$a_{10} = \frac{1}{24} = 0,041667$, $a_{20} = -\frac{1}{6} = -0,166667$, $a_{30} = \frac{1}{4} = 0,25$, $a_{13} = -\frac{1}{6} = -0,166667$, $a_{20} = \frac{1}{24} = 0,041667$; $p(t) = 0,041667 - 0,166667e^{-t} + 0,25e^{-2t} - 0,166667e^{-3t} + 0,041667e^{-4t}$; $p_{6,4,1,3}^{(1)}(t) = (0,041667 - 0,166667 e^{-\lambda t} + 0,25 e^{-2\lambda t} - 0,166667 e^{-3\lambda t} + 0,041667 e^{-4\lambda t}) e^{-3\lambda t}$
2	$a_{10} = \frac{1}{48} = 0,020833$, $a_{20} = \frac{1}{16} = 0,0625$, $a_{30} = -\frac{1}{48} = -0,020833$, $a_{13} = \frac{1}{240} = 0,004167$, $a_{20} = -\frac{1}{15} = -0,066667$; $p(t) = 0,020833 + 0,0625e^{-2t} - 0,020833e^{-4t} + 0,004167e^{-6t} - 0,066667e^{-t}$; $p_{6,4,1,3}^{(2)}(t) = (0,166667 + 0,5e^{-2\lambda t} - 0,166667 e^{-4\lambda t} + 0,033333 e^{-6\lambda t} - 0,533333 e^{-\lambda t}) e^{-6\lambda t}$
3	$a_{10} = 0,2$, $a_{11} = -0,04$, $a_{12} = 0,008$, $a_{13} = -0,0016$, $a_{20} = 0,0016$; $p(t) = 0,2 \frac{t^3}{3!} - 0,04 \frac{t^2}{2!} + 0,008t - 0,0016 + 0,0016e^{-5t}$; $p_{6,4,1,3}^{(3)}(t) = (0,48(\lambda t)^3 - 0,288(\lambda t)^2 + 0,1152 \lambda t - 0,02304 + 0,02304 e^{-5\lambda t}) e^{-12\lambda t}$

Графики трех полученных функций для интенсивности отказов $\lambda = 0,1 \text{ час}^{-1}$ изображены на рис.2.

Кривая 1 соответствует функции $p_{6,4,1,3}^{(1)}(t)$, кривая 2 – функции $p_{6,4,1,3}^{(2)}(t)$, а кривая 3 – функции $p_{6,4,1,3}^{(3)}(t)$. Из графиков видно, как увеличение нагрузки на элементы влияет на вероятность одного и того же состояния системы.

Среднее время пребывания системы в состоянии 6,4,1,3 определим по формуле

$$\tau_{6,4,1,3} = \frac{1}{\lambda} \int_0^{\infty} p_{6,4,1,3} \left(\frac{t}{\lambda} \right) dt = \frac{k_1 k_2 k_3}{\lambda c_0 (c_0 + c_4) (c_0 + c_4 + c_3) (c_0 + c_4 + c_3 + c_2) (c_0 + c_4 + c_3 + c_2 + c_1)}.$$



Результаты расчетов содержатся в табл.3.

Т а б л и ц а 3

Случай	Нагрузка				Среднее время пребывания $\tau_{6,4,1,3}$, час
	k_1	k_2	k_3	k_4	
1	1	1	1	1	0,003968
2	1	2	2	2	0,001984
3	2	2,4	3	4	0,000408

Из таблицы также следует большое отличие наработки на отказ в зависимости от степени нагрузки на элементы. Нарботка на отказ для первого случая в 2 раза больше, чем для второго, а наработка на отказ для второго случая почти в 5 раз больше, чем для третьего случая.

Состояние отказа элементов с номерами 6,4,1,3 является состоянием исправной работы системы (это есть предотказовое состояние). Вероятность безотказной работы системы определяется суммированием вероятностей всех исправных состояний. Для системы с зависимыми элементами необходимо осуществить перебор всех исправных состояний системы, вычислить вероятности этих состояний и определить сумму полученных вероятностей.

Проведенный анализ позволяет сделать следующий вывод. Расчеты характеристик надежности систем без учета зависимости элементов вслед-

ствие изменения нагрузки могут приводить к большим ошибкам. Как видно из примера, это справедливо даже для одного состояния системы, количество которых в сложной системе весьма велико. Безусловно, что величина ошибки увеличится многократно при рассмотрении всех состояний. Во избежание этого необходим обязательный учет реальных условий эксплуатации систем.

Библиографический список

1. Половко А.М., Гулов С.В. Основы теории надежности. Санкт-Петербург, БХВ-Петербург, 2006, 560 с.

И.В. Терентьев, доктор технических наук, профессор

ОЦЕНКА ПОГРЕШНОСТИ ПРОГНОЗИРОВАНИЯ ПОГОДЫ НЕ ИНСТРУМЕНТАЛЬНЫМИ МЕТОДАМИ

Для обеспечения определения прогноза с помощью разработанных методов, алгоритмов и реализующих их программ, ежедневно в 16 часов вводятся субъективные оценки состояния погоды, при этом программы позволяют удерживать в памяти 999 значений оценок. В зимние месяцы при оценке по трем градациям считается, что теплая погода (температура воздуха больше нуля) соответствует значению оценки равной 1. Прохладная погода (температура воздуха меньше нуля, но выше -10 градусов) соответствует значению оценки равной 0, холодная погода (значение оценки равно -1) соответствует температуре воздуха ниже -10 градусов, либо соответствует более высокой температуре, но при наличии сильного ветра.

Для летних месяцев принята следующая градация: считается, что теплая погода (оценка равна 1) - это солнечная погода; прохладной погоде (оценка равна 0) соответствует отсутствие солнца, а холодной погоде (оценка равна -1) соответствует наличие дождя (возможно, с ветром) либо аномально низкая температура. В весенние и осенние месяцы оценки для состояния погоды определяются аналогично летним месяцам, но с поправкой на среднегодовые температуры для этих сезонов.

Ниже приводится пример прогноза погоды, рассчитанный 26.10.2008 на 56 дней вперед для периода с 27.10.2008 – 21.12.2008.

Расчет в примере проводился для следующих методов [1 - 5].

- 1 – Алгоритм типа скользящего среднего.
- 2 – Экстраполяция с помощью кубического сплайна.
- 3 – Алгоритм оптимальной экстраполяции.

4 – Алгоритм прогнозирования с помощью неадаптивного нейрона.
 5 – Алгоритм прогнозирования с помощью двухслойной адаптивной нейронной сети.

Результаты расчетов сведены в таблицы.

В строке MAX содержится состояние, соответствующее выбору по максимальной вероятности.

Строка REAL соответствует реализовавшемуся состоянию. Доля правильных решений (определяемых по строкам REAL и MAX в скользящем окне длиной 1, 2, 3, ..., 56 дней) содержится в строке PART.

В таблицах содержатся также средние за неделю оценки, определяемые по соответствующим средним вероятностям. Ниже приведены результаты расчетов для двух методов.

1. Доля правильных решений для алгоритма типа скользящего среднего

Т а б л и ц а 1

Алгор. 1	27.10.08	28.10.08	29.10.08	30.10.08	31.10.08	1.11.08
MAX	1	1	1	1	1	1
REAL	1	1	1	1	1	1
PART	1	1	1	1	1	1

Т а б л и ц а 2

Алгор. 1	2.11.08	3.11.08	4.11.08	5.11.08	6.11.08	7.11.08
MAX	1	1	1	1	1	1
REAL	1	1	1	1	0	0
PART	1	1	1	1	0,91	0,84

Т а б л и ц а 3

Алгор. 1	8.11.08	9.11.08	10.11.08	11.11.08	12.11.08	13.11.08
MAX	0	0	1	1	1	0
REAL	0	0	1	1	1	1
PART	0,85	0,86	0,87	0,87	0,88	0,83

Т а б л и ц а 4

Алгор. 1	14.11.08	15.11.08	16.11.08	17.11.08	18.11.08	19.11.08
MAX	1	1	1	1	1	0
REAL	1	1	1	1	1	0
PART	0,84	0,85	0,86	0,86	0,87	0,87

Т а б л и ц а 5

Алгор. 1	20.11.08	21.11.08	22.11.08	23.11.08	24.11.08	25.11.08
MAX	1	1	0	0	1	1
REAL	1	1	0	1	0	1
PART	0,88	0,88	0,89	0,86	0,83	0,83

Т а б л и ц а 6

Алгор. 1	26.11.08	27.11.08	28.11.08	29.11.08	30.11.08	1.12.08
MAX	1	1	1	1	1	0
REAL	1	1	1	1	1	1
PART	0,84	0,84	0,85	0,85	0,86	0,83

Т а б л и ц а 7

Алгор. 1	2.12.08	3.12.08	4.12.08	5.12.08	6.12.08	7.12.08
MAX	0	1	1	0	1	1
REAL	1	1	1	1	1	1
PART	0,81	0,82	0,82	0,80	0,80	0,81

Т а б л и ц а 8

Алгор. 1	8.12.08	9.12.08	10.12.08	11.12.08	12.12.08	13.12.08
MAX	0	1	1	1	1	1
REAL	0	0	0	1	1	1
PART	0,81	0,80	0,78	0,78	0,79	0,79

Т а б л и ц а 9

Алгор. 1	14.12.08	15.12.08	16.12.08	17.12.08	18.12.08	19.12.08
MAX	0	1	1	1	1	0
REAL	0	0	0	0	0	0
PART	0,80	0,78	0,76	0,75	0,74	0,74

Т а б л и ц а 10

Алгор. 1	20.12.08	21.12.08
MAX	0	1
REAL	0	0
PART	0,75	0,73

Т а б л и ц а 11

Алгор. 1	27.10.08- 2.11.08	3.11.08- 9.11.08	10.11.08- 16.11.08	17.11.08- 23.11.08
MAX	1	1	1	1
REAL	1	0	1	1
PART	1	0,5	0,67	0,75

Т а б л и ц а 12

Алгор. 1	24.11.08- 30.11.08	1.12.08- 7.12.08	8.12.08- 14.12.08	15.12.08- 21.12.08
MAX	1	1	1	1
REAL	1	1	0	0
PART	0,80	0,83	0,71	0,63

2. Доля правильных решений при прогнозировании с помощью оптимальной экстраполяции

Т а б л и ц а 13

Алгор. 3	27.10.08	28.10.08	29.10.08	30.10.08	31.10.08	1.11.08
MAX	1	1	1	1	0	1
REAL	1	1	1	1	1	1
PART	1	1	1	1	0,80	0,83

Т а б л и ц а 14

Алгор. 3	2.11.08	3.11.08	4.11.08	5.11.08	6.11.08	7.11.08
MAX	0	0	0	0	1	0
REAL	1	1	1	1	0	0
PART	0,71	0,62	0,56	0,50	0,45	0,50

Т а б л и ц а 15

Алгор. 3	8.11.08	9.11.08	10.11.08	11.11.08	12.11.08	13.11.08
MAX	0	0	1	0	1	0
REAL	0	0	1	1	1	1
PART	0,54	0,57	0,60	0,56	0,59	0,56

Т а б л и ц а 16

Алгор. 3	14.11.08	15.11.08	16.11.08	17.11.08	18.11.08	19.11.08
MAX	1	0	1	1	1	0
REAL	1	1	1	1	1	0
PART	0,58	0,55	0,57	0,59	0,61	0,62

Т а б л и ц а 17

Алгор. 3	20.11.08	21.11.08	22.11.08	23.11.08	24.11.08	25.11.08
MAX	1	1	0	0	1	1
REAL	1	1	0	1	0	1
PART	0,64	0,65	0,67	0,64	0,60	0,63

Т а б л и ц а 18

Алгор. 3	26.11.08	27.11.08	28.11.08	29.11.08	30.11.08	1.12.08
MAX	1	0	1	1	1	1
REAL	1	1	1	1	1	1
PART	0,65	0,62	0,64	0,65	0,66	0,67

Т а б л и ц а 19

Алгор. 3	2.12.08	3.12.08	4.12.08	5.12.08	6.12.08	7.12.08
MAX	0	1	1	0	1	1
REAL	1	1	1	1	1	1
PART	0,65	0,66	0,67	0,65	0,66	0,67

Т а б л и ц а 20

Алгор. 3	8.12.08	9.12.08	10.12.08	11.12.08	12.12.08	13.12.08
MAX	0	1	1	1	1	0
REAL	0	0	0	1	1	1
PART	0,67	0,66	0,64	0,65	0,66	0,64

Т а б л и ц а 21

Алгор. 3	14.12.08	15.12.08	16.12.08	17.12.08	18.12.08	19.12.08
MAX	1	1	1	1	1	0
REAL	0	0	0	0	0	0
PART	0,63	0,62	0,61	0,60	0,58	0,59

Т а б л и ц а 22

Алгор. 3	20.12.08	21.12.08
MAX	0	1
REAL	0	0
PART	0,67	0,66

Т а б л и ц а 23

Алгор. 3	27.10.08- 2.11.08	3.11.08- 9.11.08	10.11.08- 16.11.08	17.11.08- 23.11.08
MAX	1	0	1	1
REAL	1	0	1	1
PART	1	1	1	1

Т а б л и ц а 24

Алгор. 3	24.11.08- 30.11.08	1.12.08- 7.12.08	8.12.08- 14.12.08	15.12.08- 21.12.08
MAX	1	1	1	1
REAL	1	1	0	0
PART	1	1	0,86	0,75

3. Выводы по результатам расчета

Обоснованный выбор алгоритмов, обеспечивающих достижение минимальной вероятности ошибки прогноза, определяется на некотором множестве расчетов. При этом оценивается доля правильных решений для каждого расчета, которая затем усредняется по множеству расчетов - для определения оценки вероятности правильного решения. В примере расчета доля правильных решений определяется по строкам REAL и MAX таблиц в скользящем окне и содержится в строке PART.

Поясним методику отбора наилучших алгоритмов на примере отбора по доле правильных решений приведенного расчета, содержащихся в стро-

ке PART таблиц. При сравнении алгоритмов прогнозирования на каждый день по значению доли правильных решений наилучшим оказался алгоритм номер 1 - алгоритм типа скользящего среднего (см. таблицы 1 – 10, 13 – 22). При сравнении алгоритмов прогнозирования средних значений за неделю наилучшими оказались алгоритмы оптимальной экстраполяции и алгоритм прогнозирования с помощью неадаптивного нейрона, которые показали одинаковые результаты. Прогнозирование погоды на каждый день методом голосования по трем и пяти методам оказалось хуже по сравнению с алгоритмом 1, так как методы существенно различаются по значению доли правильных решений. Однако прогнозирование средних значений погоды за неделю с помощью голосования по 5 методам показало совпадающие с алгоритмом оптимальной экстраполяции и алгоритмом прогнозирования с помощью неадаптивного нейрона результаты.

Заключение

1. При прогнозировании погоды на каждый день доля правильных решений меняется от значения близкого к 1 (для нескольких дней) до 0,7 (для нескольких десятков дней) - см. таблицы 1 – 10, 13 - 22.

2. При прогнозировании средних значений погоды за неделю доля правильных решений меняется от значения близкого к 1 (для примерно 6 недель) до 0,7 – 0,8 для 8 недель – см. таблицы 11, 12, 23, 24.

3. Практически достоверное прогнозирование средних значений погоды за неделю (для примерно шести недель) позволяет уверенно планировать лесохозяйственные работы на соответствующие сроки и повысить их эффективность. Оценка достоверности прогнозирования позволяет выбрать повторяемость прогнозов: 1 – 2 раза в неделю для прогнозов на каждый день и 1 раз в месяц при прогнозировании средних значений погоды за неделю.

Приведем пример использования средних за неделю значений погоды. Так, например, при прогнозировании средних значений на 6 недель вперед шестая неделя по прогнозу оказалась теплой (то есть по крайней мере 4 дня этой недели по прогнозу будут теплыми). Однако первые три дня шестой недели выпали прохладными. С вероятностью близкой к единице можно утверждать, что последующие четыре дня шестой недели будут теплыми и благоприятными для проведения работ на открытом воздухе.

4. Организация не инструментальных измерений в нескольких населенных пунктах позволяет учесть пространственно-временной характер метеорологических процессов и уменьшить вероятность появления ошибочных прогнозов.

5. Не инструментальный прогноз погоды для Петербурга можно найти в Интернете по адресу <http://igo23325045.narod.ru/> .

Библиографический список

1. *Терентьев И.В.* Не инструментальное прогнозирование погоды на компьютере // Тр. VIII-й Междунар. науч.-практ. конф. «Системный анализ в проектировании и управлении»(22–24 июня 2004). СПб.: СПбГПУ. Ч. 2. С. 164–167.
2. *Терентьев И.В.* Определение апостериорных вероятностей состояний процесса в будущем по его не инструментальным оценкам в прошлом // Тр. VI-й Междунар. науч.-технич. конф. «Компьютерное моделирование 2005» (28 июня – 2 июля 2005). СПб.: СПбГПУ. С. 225–230.
3. *Терентьев И.В.* Определение состояния процесса в будущем по его не инструментальным оценкам в прошлом на основе закона больших чисел // Тр. X-й Междунар. науч.-практич. конф. «Системный анализ в проектировании и управлении» (28 июня – 10 июля 2006). СПб.: СПбГПУ. Ч. 1. С. 179–183.
4. *Терентьев И.В.* Определение состояния процесса в будущем по его не инструментальным оценкам в прошлом с помощью оптимальной экстраполяции и интерполяции // Тр. XI-й Междунар. науч.-практ. конф. «Системный анализ в проектировании и управлении» (28 июня–30 июня 2007). СПб.: СПбГПУ. Ч. 1. С. 132–141.
5. *Терентьев И.В.* Определение состояния процесса в будущем по его не инструментальным оценкам в прошлом с помощью неадаптивной и адаптивной нейронной сети // Тр. XII-й Междунар. науч.-практич. конф. «Системный анализ в проектировании и управлении» (24 июня – 26 июня 2008). СПб.: СПбГПУ. Ч. 2. С. 119–128.

Н.П. Васильев, кандидат технических наук, доцент
А.М. Заяц, кандидат технических наук, профессор

ОПЫТ ИСПОЛЬЗОВАНИЯ AJAX ТЕХНОЛОГИЙ

В основе концепции построения комплекса учебных виртуальных лесничеств (УВЛ) положены современные INTERNET\INTRANET технологии, предполагающие их организацию УВЛ как систему хранения, передачи, обработки и доступа к внутриакадемической информации и информации других организаций с использованием средств локальных сетей и Интернет [2,3].

Эта организация, на основе гипертекста, позволяют создать информационную инфраструктуру учебных виртуальных лесничеств, объединив различные информационные ресурсы и предоставив к ним единообразный доступ.

Работа пользователя «внутри» учебного виртуального лесничества в большинстве своем сводится к поиску, разработке, заполнению и редактированию документов, представленных в различных форматах, что приводит к частому обращению браузера к Web-серверу для загрузки страниц, особенно если в активной среде УВЛ находится несколько сотен пользователей.

Анализ показывает, что пользователем обычно производятся операции с отдельными объектами страницы, при которых полная перезагрузка страницы не требуется. Например, необходимо обновить таблицу, которая является только частью страницы, данными, содержащимися на сервере, или обновить меню комбинированного списка (combobox) в процессе его заполнения пользователем, или динамически подгрузить ветку некоторой древовидной структуры, при этом перезагрузка страницы нежелательна, так как растет трафик и увеличивается время отклика (реакции) системы.

Одним из эффективных подходов, решающим эту задачу является технология **AJAX** (*Asynchronous Javascript and XML*), используемая для построения интерактивных пользовательских интерфейсов веб – приложений [1]. Данная технология построена на принципе выполнения запроса к серверу с использованием JavaScript и получению результата опять же, с помощью JavaScript, что позволяет избежать перегрузки страницы и, следовательно, имеет несколько неоспоримых преимуществ:

1. На сервер отправляются не все элементы страницы (точнее не их значения), а только те минимальные данные, которые необходимы для выполнения того или иного запроса и в ответ принимается не вся страница, а только необходимые данные, что позволяет уменьшить трафик в десятки (а иногда и в сотни) раз.

2. Не происходит перегрузка страницы в браузере и у пользователя создается впечатление, что все происходит на его компьютере.

Возникает вопрос: как же реализуется запрос к серверу без перегрузки страницы, фактически минуя услуги браузера? Ответ: посредством объекта ActiveX, который поставляется с браузером. Этот объект чаще всего называют XMLHttpRequest, но в реестре он, конечно, «прописан» под другим именем. Достать его позволяет JavaScript, который имеет возможность работать с ActiveX компонентами. Вот простой кроссбраузерный код, реализующий эту возможность:

```
function getXmlHttp() {
    var xmlhttp = null;
    try { xmlhttp = new ActiveXObject("Msxml2.XMLHTTP"); }
    catch(e) {
        try { xmlhttp = new ActiveXObject("Microsoft.XMLHTTP"); }
        catch(e) {}
    }
    if (!xmlhttp && typeof XMLHttpRequest!='undefined') {
        xmlhttp = new XMLHttpRequest();
    }
    return xmlhttp;
}
```


Функция перебирает возможные внутренние реализации и возвращает начальный объект XMLHttpRequest. Существует и масса других рабочих кроссбраузерных функций, однако все они, по сути, делают то же самое.

Возможны два варианта использования полученного таким образом компонента: синхронный и асинхронный. Первый вариант практически не используется, поскольку в случае разрыва соединения по тем или иным причинам, приводит к «зависанию» браузера, но для полноты картины приведем пример синхронной загрузки:

```
var xmlhttp = getXmlHttpRequest()
xmlhttp.open('GET', '/tst/test.html', false);
xmlhttp.send(null);
if(xmlhttp.status == 200) {
    alert(xmlhttp.responseText);
}
```

Здесь сначала создается запрос, задается открытие (open) синхронного соединения с адресом /tst/test.html и запрос отправляется с null, т.е. без данных: send(null).

При синхронном запросе браузер "подвисает" и ждет на строчке 3, пока сервер не ответит на запрос. Когда ответ получен - выполняется строка 4, код ответа сравнивается с 200 (ОК), и при помощи alert печатается текст ответа сервера. Все максимально просто.

Следующий пример делает то же самое, но асинхронно, т. е. браузер не ждет выполнения запроса для продолжения скрипта. Вместо этого к свойству onreadystatechange «подвешивается» функция, которую запрос вызовет сам, когда получит ответ с сервера:

```
var xmlhttp = getXmlHttpRequest()
xmlhttp.open('GET', '/xhr/test.html', true);
xmlhttp.onreadystatechange = function() {
    if (xmlhttp.readyState == 4) {
        if(xmlhttp.status == 200) {
            alert(xmlhttp.responseText);
        }
    }
};
xmlhttp.send(null);
```

Асинхронность включается третьим параметром функции open. В отличие от синхронного запроса, функция send() не останавливает (принято говорить: не блокирует) выполнение скрипта, а просто отправляет запрос.

Разработки в AJAX привели к созданию библиотек JavaScript, получивших название фреймворков (framework), которые в значительной степени упростили использование этой технологии, и разработку WEB интерфейсов в целом. К числу популярных фреймворков можно отнести:

jQuery (http://jquery.com/),	SWFObject
jQuery UI (http://ui.jquery.com/),	(http://code.google.com/p/swfobject/),
Prototype	Yahoo! User Interface Library (YUI)
(http://www.prototypejs.org/),	(http://developer.yahoo.com/),
script.aculo.us	наконец, ExtJS (http://extjs.com/).
(http://script.aculo.us/),	
MooTools (http://mootools.net/),	
Dojo (http://dojotoolkit.org/),	

Использование фреймворков, практически полностью освобождает WEB-программиста от знания всех внутренних нюансов AJAX. Можно отметить, что фреймворки инкапсулируют все детали этой технологии, а оставляют на поверхности только необходимые рычаги управления. Вот так, например, можно реализовать AJAX запрос к серверу с помощью ExtJS:

```
Ext.Ajax.request({
    url: 'ajax_test.php',
    params: { action: 'getDate' },
    method: 'GET',
    success: function( result, request ) {
        Ext.MessageBox.alert('Success', 'Data return from the
server: '+
    result.responseText);
    },
    failure: function( result, request) {
        Ext.MessageBox.alert('Failed', result.responseText);
    }
});
```

Здесь в json-нотации задаются все необходимые моменты реализации асинхронного AJAX запроса. Это: что должно отработать на сервере (url: 'ajax_test.php'), параметры, которые будут переданы в данном случае - ajax_test.php, метод http запроса (GET или POST), наконец, что требуется сделать в случае успешного завершения общения с сервером (success: ...), или в случае неудачи (failure: ...). Более того, фреймворки, в том числе и

ExtJS, для контейнеров (типа div и др.) поддерживают методы типа load (reload), которые вообще освобождают программиста от отслеживания деталей AJAX.

Таким образом, технология AJAX совместно с фреймворками позволяют улучшить функциональность WEB интерфейсов за счет возможности непосредственного обращения к серверу из кода JavaScript. Общение с сервером осуществляется через http протокол.

Программирование AJAX сложнее обычного программирования в JavaScript, поскольку является асинхронным. Это означает, что требуется самостоятельно отслеживать момент получения ответа от сервера, связанные с этим таймауты, и возможные нарушения соединения. Однако, большинство существующих в настоящее время фреймворков берут на себя всю низкоуровневую обработку указанных деталей, делая доступным и достаточно простым программирование AJAX.

Технологии AJAX не только улучшают дизайн, но и значительно ускоряют работу WEB интерфейсов, поскольку позволяют осуществлять загрузку (перезагрузку) только требуемых в данный момент частей страницы. Однако за это приходится расплачиваться загрузкой достаточно объемного JavaScript кода используемого фреймворка (если, конечно, он используется).

Технология AJAX также имеет свои ограничения. Например, эта технология не позволяет выполнить загрузку (upload) файлов на сервер <http://xmlhttprequest.ru/>

Библиографический список

1. Заяц А.М. Новый подход в практической подготовке студентов Лесотехнической академии//Сборник научных трудов "Информационные системы и технологии: теория и практика". -СПб.: СПбГЛТА, 2008, 4 - 13 с.
2. Заяц А.М. Виртуальное лесничество подготовит грамотные кадры // ЛЕСПРОМинформ. 2008. № 2, 26-28 с.

Н.П. Васильев, кандидат технических наук, доцент

МОДУЛЬНОСТЬ И УНИВЕРСАЛЬНОСТЬ – КЛЮЧИ К БЫСТРОЙ РАЗРАБОТКЕ WEB ИНТЕРФЕЙСОВ ИНФОРМАЦИОННЫХ СИСТЕМ

Каждый разработчик информационных систем, приступая к новому проекту, сознательно или подсознательно использует уже накопленные у него подходы и решения из своего прошлого опыта. Эффективность разработок напрямую зависит от степени их универсальности. Если повезет, то может оказаться, что некоторые решения «один в один» ложатся под новые задачи, но такое случается редко. Как правило, требуются существенные или несущественные изменения кода, которые в свою очередь ведут к повторной проработке и осознанию того, что уже сделано. С одной стороны, это полезно, поскольку позволяет усовершенствовать код, выявить скрытые ошибки узкие места, а с другой стороны, часто приводит к нелепым и обидным ошибкам, поиск и исправление которых отнимает массу времени. Более того, такие модифицированные решения похожих задач кочуют из проекта в проект, дублируют код, дублируют старые и привносят новые ошибки, так что порой сам автор путается в многочисленных версиях.

Выход один – создание унифицированных, легко настраиваемых модулей, из которых, как из кирпичиков можно сложить решение схожих задач. Принципы следующие: универсальность и возможность легкой и понятной настройки, модульность, исключение дублирования кода, от многочисленных версий решения приблизительно одних и тех же задач к отлаженным универсальным модулям, настраиваемым под нужды и особенности конкретных ситуаций.

Конечно, ничего не бывает бесплатным: за универсальность придется расплачиваться более объемным кодом, более глубокой проработкой архитектуры универсальных решений, а также риском получения от заказчика дополнительных требований, которые, к сожалению, не вписываются в эту универсальность – и все опять пойдет по кругу. Взамен - отлаженные в многочисленных проектах модули, надежность создаваемого программного обеспечения, быстрота разработки. Сказанное не является чем-то новым. Указанные принципы давно заложены в основу таких систем быстрой разработки программного обеспечения, как Delphi или C++Builder. В этих системах несколькими движениями мыши можно поместить в форму тот или иной компонент, настроить его свойства, и вот уже готова, например, полнофункциональная сетка для просмотра содержимого базы данных.

Однако, что касается разработки WEB интерфейсов, то здесь по-прежнему в основном используется «голый» PHP или Perl на стороне сер-

вера и JavaScript на стороне клиента. Правда, в последнее время появились универсальные библиотеки на JavaScript, получившие названия Фреймворков, широко использующие AJAX технологии, предназначенные для быстрой и профессиональной разработки клиентской части приложений.

Универсальный модуль для работы с таблицами. Ядром любой информационной системы является база данных, управляемая, как правило, тем или иным SQL сервером. Поэтому первоочередной задачей разработчика, после разработки продуманной схемы отношений, адекватно представляющей предметную область, является разработка удобного интерфейса по ведению таблиц базы. Что здесь имеется в виду под ведением таблиц базы. Это, конечно же, возможность просмотра, добавления и корректировки данных. Это возможность динамической фильтрации данных. Это возможность представления содержимого связанных таблиц.

Программисту при этом приходится переписывать приблизительно одинаковые html-формы и связанный с этими формами JavaScript-код на стороне клиента, и приблизительно одинаковые PHP скрипты по обработке данных этих форм на стороне сервера. Каждый раз приходится решать ряд похожих, повторяющихся из проекта в проект задач. Вот более подробный первоочередной перечень этих задач: прежде всего «красивое» представление содержимого таблицы (или связанных таблиц), которое предполагает возможность сортировки по данным в том или ином столбце, возможность задания фильтров для данных в ряде столбцов, наконец, возможность выбора для визуализации тех или иных столбцов таблицы. К этому перечню также относятся задачи добавления, корректировки и удаления строк таблицы. Добавление и корректировка требуют, в свою очередь, генерации форм по вводу данных и выполнения контроля введенных данных на стороне клиента. На стороне сервера потребуются PHP скрипты для добавления данных формы в таблицу, или корректировки данных таблицы в соответствии с данными формы, наконец, самое простое, удаления записей в таблице. Еще раз следует подчеркнуть, что это ряд однотипных задач, которые по своей сути не отличаются для различных таблиц, или различных проектов. Естественно, возникает вопрос, а нельзя ли унифицировать эти задачи, и написать единый настраиваемый модуль для их решения, о чем и шла речь выше.

Автором была предпринята попытка реализации такого модуля. Суть решения состоит в следующем: сосредоточить все необходимые описания в одном месте, например, в текстовом файле, в определенном формате, который будет использоваться из различных точек модуля. Почему в текстовом виде, а не в каких-либо настроенных таблицах базы. Просто текстовый формат более доступен для корректировки, чем таблицы базы, для корректировки которых, собственно, и предусмотрен модуль.

Связующим звеном для всех составляющих модуля является класс TABLE, реализованный на PHP, на стороне сервера, на входе (в конструкторе) которого указывается упомянутый файл с настроечными описаниями. В конструкторе выполняется разбор описаний и соответствующая этим описаниям инициализация переменных экземпляра класса. Каждая составляющая модуля включает указанный класс, и, таким образом, работа любой составляющей модуля начинается с настроечной части.

Работа с таблицей, прежде всего, предполагает генерацию ее HTML представления. Для этого требуется, как минимум, знать имя таблицы в базе, набор полей, которые требуется отобразить, вразумительные названия колонок под эти поля, а также как представлять значения выбранных полей (то есть формат представления). По этим данным составляется SQL запрос для выборки данных, и выбранные данные представляются в HTML формате, в требуемом виде.

Для генерации формы добавления данных в таблицу опять же потребуются те же самые исходные настроечные данные. Форма корректировки строки таблицы ничем не отличается в этом плане от формы добавления. Таким образом, все составляющие модуля замыкаются на общий класс TABLE, где и сосредоточены все требуемые методы. В то же время все настроечные данные доступны в простом текстовом виде. Вот примерное содержимое настроечного файла:

```
# название таблицы
table_title:      Здесь даем название таблицы

# имя таблицы в базе данных
table:           tbl

# поля в таблице, указанные через разделитель |
fields:          field1|field2|...

# значения по умолчанию для некоторых полей через разделитель |
# фактически это фильтр, то есть из таблицы выбираются строки
ТОЛЬКО
# с заданными значениями указанных полей
fields values:   fld=v1|fld=v1 ...

# ключевое поле таблицы
key:             key_fld
# названия колонок таблицы (для каждого поля из строки fields)
titles:          col_ttl1|col_ttl2 ...

# какие колонки показывать при начальной загрузке таблицы
show:           1|1|0 ...
```

```

# типы колонок, то есть как представлять содержимое колонок
# таблицы (auto, int, number, string, text, date, enum, myenum, foreign)
types:          string|foreign|enum; opt1,opt2|string|int ...

# обязательные поля
obligatory:    1|1|1|0 ...

# по каким полям выполнить сортировку при начальной загрузке
таблицы
order by:      field1, field2

# можно отфильтровать записи и так
filter:       фильтр, допустимый в предложении WHERE SQL запро-
са

# размеры формы редактирования (добавления)
form size:    500:300

# место под метки полей формы редактирования-(добавления)
label width:  150

```

В данном описании решеткой отделены комментарии. Если в таблице присутствуют ссылки на другую таблицу, так называемые `foreign keys`, то такие связанные таблицы также можно представить, только соответствующие поля в строке `fields` требуется раскрыть по схеме: `field;foreign_table;key;showfield`

Здесь: `foreign_table` – таблица, на ключевое поле которой ссылается поле `field`, `key` – имя ключевого поля таблицы `foreign_table`, `showfield` – поле таблицы `foreign_table`, которое требуется отобразить в соответствующей колонке.

В строке `types` указывается форма представления значений соответствующих полей, а также способ их ввода в формах добавления (редактирования). Так, например, если указан тип `enum` (т.е. перечисление), то это означает, что в данном поле может находиться ряд фиксированных значений, которые перечисляются при задании типа поля через запятую, и, следовательно, в форме добавления редактирования для них можно использовать поле ввода типа `select` с соответствующим набором опций. Аналогичен тип `foreign`, который отличается от `enum` способом формирования опций. Для типа `foreign` возможные значения поля выбираются из связанной таблицы, которая должна быть указана в соответствующем описании поля (строка `fields`). Описания типов используются также и для формирования динамических фильтров. Назначения остальных строк конфигурационного файла понятны без комментариев.

Заключение. Модульность и универсальность – основные принципы, на которых строится надежная и быстрая разработка программного обеспечения. Эти принципы сохраняются и для WEB программирования. В настоящее время существует масса так называемых движков информационных систем, распространяемых на коммерческой и свободной основе. Но подобные разработки даже с очень гибкой системой настройки вряд ли можно считать средствами программирования для создания WEB интерфейсов информационных систем. Альтернативой может быть разработка универсальных настраиваемых модулей, из которых можно собрать требуемое решение той или иной информационной задачи. Под модулем здесь понимается весь необходимый набор файлов, располагаемый на серверной и клиентской стороне, который связан воедино для универсального решения ряда однотипных задач. Автором был разработан и опробован подобный модуль для представления и корректировки таблиц базы данных. Весь спектр задач, который решается при этом управляется с помощью единого класса TABLE и настраивается из одного конфигурационного файла.

И.В. Ганичев, кандидат технических наук, доцент
Е. Г. Троицкая, А. Г. Троицкая, студенты

СТРАТЫ В МОДЕЛЯХ ЭВМ

Введение. В информационных системах (ИС) выделяют подсистему обработки информации, которая состоит из ЭВМ и их комплексов. Анализ построения и функционирования сложных систем неизбежно приводит к иерархическим моделям в форме страт. В известных моделях ЭВМ [1, 2] отражается отношение «аппаратура – программа», но «алгоритмическая» составляющая (важная для реализации прикладных программ) вообще не объявлена.

Цель работы – максимально широкое применение подхода стратификации и обозначение новых уровней иерархии с выявлением аппаратной, программной и алгоритмической составляющих (Hard-, Soft-, Brain ware) в объектах исследования – стратах ЭВМ.

Состав и детализация страт зависит от целей их формирования, интересов исследователя, его опыта и квалификации. Так в [3] выделено описание страт в цепочке «алгоритм-программа-команда».

Аналитическая модель фон Неймана, сформулировавшего принципы построения ЭВМ, обязательно должна входить в базовый набор страт. Включение данной вербальной страты, должным образом сформулирован-

ной, позволяет открыть в учебном процессе путь к синтезу функциональных структур ЭВМ, соответствующих архитектуре «машины фон Неймана».

Таким образом, строим первую цепочку страт, дополняя уровни {1.. 3} иерархии из модели Майорова [2] очередной по старшинству стратой фон Неймана, модель которого часто называют машиной Неймана.



Рис. Результаты стратификации

Программно-алгоритмическая составляющая образует уровни второй цепочки (см. правую часть рис.).

Перед нами встали задачи:

- выявить макро-модель (высшую страту), которая бы показала потенциальные возможности и философию системы,
- сформировать иерархию так, чтобы наглядно была видна взаимосвязь и единство всех важных компонентов системы.

В качестве высшей страты, которая отвечает на поставленные вопросы, может быть взята абстрактная машина Тьюринга. Безусловное, алгоритмическое начало абстрактного построения Тьюринга связано с конструктивными решениями, в которых виден прообраз ВМ. Можно выявить подсистемы – память в виде ленты с ячейками, в которые вводятся символы входного алфавита, управляющий автомат с выделенными состояниями, исполнительное устройство, способное читать ленту и печатать при необходимости новые символы. Для абстрактного автомата составляются программы. Программирование абстрактной машины Тьюринга использует основные алгоритмические конструкции – разветвление и циклы.

Представленная многоуровневая модель ЭВМ позволяет выявить разновидности ПО (в этом плане модель является полезной), а также узловые точки противоречий между концепциями построения подсистем на верхних уровнях и реализациями соответствующих компонентов на нижних

уровнях иерархии как аппаратного, так и ПО. По результатам системного анализа необходимо сформулировать меры для повышения эффективности и качества функционирования подсистем ИС. Несоответствие концепций и принципов ЭВМ принято называть в современной литературе семантическим разрывом [4].

Заявленный подход в стратификации моделей наряду с ясным пониманием, в каких областях иерархии зарождается семантический разрыв, позволит обратить внимание на актуальные направления развития архитектуры компьютеров.

Заключение. Представленная разработка может быть использована как полезная модель для создания на ее основе гипертекста по страницам ЭВМ.

Библиографический список

1. Месарович М., Мако Д., Такахара И. Теория иерархических многоуровневых систем. – М.: Мир, 1973.
2. Майоров С.А., Новиков Г.И. Структура электронных вычислительных машин Л.: Машиностроение, Ленингр. отделение, 1979.
3. Новые подходы к моделированию процессов и корпоративной архитектуры предприятия (Бизнес-инжиниринг и ОРГ-МАСТЕР®: что, зачем, как?) // Григорьев Л. Ю., Кудрявцев Д. В., Спиридонов В. В., Кислова В. В.
http://www.big.spb.ru/publications/bigspb/metodology/new_model_corp_arch.shtml
4. Буза М.К. Архитектура компьютеров. – Минск: Новое знание, 2006.

И.В. Ганичев, кандидат технических наук, доцент
Н.П. Васильев, кандидат технических наук, доцент
С.В. Киселева, доцент

JAVASCRIPT В ИНФОРМАЦИОННЫХ СИСТЕМАХ

JavaScript в настоящее время занимает достойное место среди языков Web-программирования. За последние десятилетия в связи с развитием Интернет-технологий изменились подходы, как к проектированию информационных систем (ИС), так и к программированию. Непонимание и даже неприятие со стороны пользователей, не знакомых с объектным подходом, может быть связано с трудностями его восприятия. Новые изменения информационных технологий отражаются в следующем:

- обилие новых терминов и технологий, за которыми стоят и соответствующие средства их реализации; появление в названиях префикса

«кросс», расширяющего основной термин – кроссплатформа, кроссязыки, кросспроектирование, кроссбраузерность;

- проникновение объектно-ориентированного подхода во все Web- приложения;
- облегчение представления и модификации содержания Web страниц;
- симбиоз языков Java, JavaScript и XML на стороне клиента,
- использование языков PHP, Perl, ASP и др. на стороне сервера,
- выделение «бизнес логики» в ИС и перенос выполнения части пользовательского приложения на сторону клиента.

В указанном ряду языков особое место занял язык сценариев JavaScript. JavaScript дает возможность внедрения команд процедурного языка в HTML-документ страницы, отображаемой браузером. Браузер интерпретирует внедренные команды и выполняет их.

Браузер может сообщать о множестве событий, для обработки которых составляется сценарий на языке JavaScript, согласно концепциям объектных моделей (ОМ) браузера – BOM и документа – DOM. Таким образом, JavaScript является объектным языком, и кто знает основы объектного программирования, тот легче овладевает самим языком.

Пользователь имеет дело с тремя базовыми элементами - объектами, сообщениями и классами.

Объект определяется как многократно используемый автономный программный модуль, *инкапсулирующий* две структурные части (переменные-свойства и процедуры-методы) и содержащий всю необходимую информацию для его реализации.

Хорошей иллюстрацией для восприятия системности данных категорий является объектная модель в виде треугольника Уемова, вершины которого соответственно отражают с одной стороны сами объекты, с другой – свойства-данные, а с третьей – методы, устанавливающие отношения между ними (рис.1). В скобках указаны авторские определения. Пунктирное обрамление иллюстрирует инкапсуляцию.

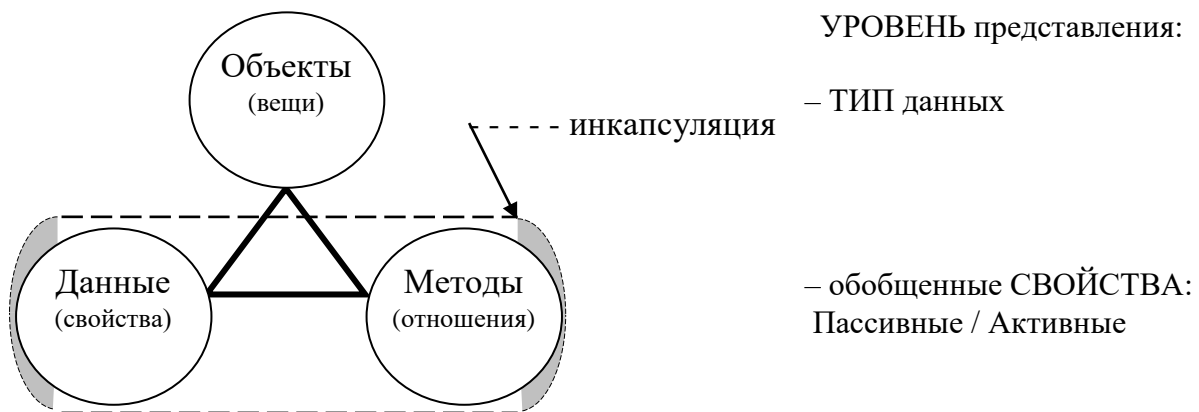


Рис. 1. Системная интерпретация объектного подхода (инкапсуляция)

Свойство (или объектная переменная) в коде программы указывается после имени объекта за разделителем, который называется синтаксической точкой, «по схеме» **объект.свойство**. Системная сложность объекта отражается возможностью включения в объектную переменную другого объекта.

Методы представляют собой набор процедур и функций, определяющих алгоритм функционирования объекта. Указание на использование метода в коде программы обязательно завершается парой скобок «по схеме» **метод()**.

На уровне представлений объект может отражать тип данных, а инкапсулированные элементы общее свойство, для значений данных – пассивные, для методов – активные.

Для связи между объектами применяются сообщения, состоящие из трех частей:

- идентификатор объекта, которому оно адресовано;
- имя метода, который должен быть исполнен в указанном объекте;
- дополнительная информация – фактических значений, придаваемым формальным параметрам с целью настройки выбранного метода.

Использование сообщений основано на системе правил взаимодействия составных частей проектируемой программной системы. Здесь остается в тени внутренняя организация объектов, ее можно варьировать при условии сохранения обновленным объектом восприимчивости к тем же сообщениям. При этом можно перестраивать многомодульные системы, сохраняя общий алгоритм их функционирования в проектах.

Класс представляет *шаблоны* для многократной генерации однотипных объектов, *содержащие* информацию только о методах и типах объектных переменных. В генерируемые объекты включается информация о значениях объектных переменных. Таким образом, объект есть экземпляр класса. Подклассы *наследуют* свойства высших классов, создают иерархическую цепочку, что позволяет использовать иерархические

структуры и облегчает проектирование сложных многомодульных программных систем.

Таким образом, абстрагирование данных, сокрытие деталей и передача свойств в подклассы определяют новую объектную методологию в следующих направлениях:

- инкапсуляция – объединение записи с процедурами и функциями, формирование нового типа данных «объекты»;
- наследование – иерархия порожденных объектов с формированием доступ каждого из них к процедурам и данным своего предка;
- полиморфизм – присвоение единого имени процедуре, передаваемой по иерархической цепочке в пределах одного класса, с выполнением ее способом, соответствующим каждому объекту иерархии.

Объемная интерпретация проектируемых объектов в виде связанных моделей Умова отражает структуру, информационные и функциональные связи ИС.

Следует заметить, что язык JavaScript «не стремился» быть полноценным языком программирования, а только обеспечивал взаимодействие пользователя с Web-страницами, придавая им динамичность и интерактивность (в настоящее время эти возможности стремительно развиваются). Язык JavaScript объектно-ориентированный, но основан на прототипах (а не на классах). Имеются четыре типа объектов: встроенные объекты языка, объекты интерпретирующей системы (браузера), объекты документа и объекты пользователя (программиста). К данным применяется слабый контроль типов.

Группа методов, которым JavaScript придает новое системное свойство, получила название «асинхронное использование JavaScript и XML» или AJAX (Asynchronous JavaScript And XML). Главная особенность этого подхода отражена первой буквой слова «асинхронность». Методы AJAX представляет одну из развивающихся ветвей Web-программирования.

AJAX представлен инфраструктурой из документов HTML, объектной модели документа DOM, каскадных таблиц стилей CSS, языка программирования JavaScript и объекта XMLHttpRequest.

Информационные технологии, развиваемые на базе JavaScript, используют для удаленного взаимодействия и контакта с серверами Интернета. Подобная идея была подхвачена многими разработчиками JavaScript сайтов для исключения прерываний интерактивного взаимодействия во время перерисовки веб-страниц при обновлении данных.

Асинхронность обеспечена незаметной для пользователя фоновой работой объекта XMLHttpRequest при отправке запроса на Web-сервер. Ответ Web-сервера раскрывается только из скриптового языка (например, JavaScript), что придает уникальные возможности для Web-программирования.

Сценарии с внешними связями на стороне клиента образуют основу доступа к извлекаемому сервером содержимому. При этом некоторая функциональная часть приложения выполняется на стороне клиента и может быть реализована с помощью средств JavaScript.

В AJAX-технологиях посредником является объект XMLHttpRequest. Данный компонент-посредник отправляет обычный запрос HTTP и дожидается (синхронно или асинхронно) завершения его обработки. Получив готовые данные ответа, посредник вызывает функцию обратного вызова JavaScript; это обычная функция, которая иницируется только после получения ответа от сервера. Объект XMLHttpRequest дает возможность сценарному коду отправлять запросы HTTP и немедленно обрабатывать полученные ответы.

Можно теперь передавать клиенту код из нескольких тысяч строк, что при нормальном подключении к Web-серверу требует немного времени с учетом выполнения его без задержек на любой современной машине.

Чем больше функций возложено на клиентскую часть приложения и чем интенсивней обмен данными с сервером, тем значительней сокращение суммарного трафика сеанса для новой технологии.

Все технологические отличия новых решений выражают в четырех позициях:

- в браузере нет периодов вынужденных ожиданий доставки очередного документа;
- браузер имеет дело с единым функциональным приложением, а не с набором самостоятельных документов;
- сервер отправляет теперь клиенту не содержимое отдельных документов, а данные;
- пользователь получил возможность непрерывного взаимодействия с приложением.

При проектировании программных продуктов важно использовать опыт разработчиков и архитектурные представления, отраженные в наборе образцов проектирования, или шаблонов типа Pattern.

Архитектура программного обеспечения, в которой модель данных приложения, пользовательский интерфейс и управляющая логика разделены на три отдельных компонента, так, что модификация одного из компонентов оказывает минимальное воздействие на другие компоненты – она представлена в форме MVC (Model–View–Controller или «Модель–представление–контроллер»).

Шаблон MVC позволяет разделить данные, представление и обработку действий пользователя на три отдельных компонента:

- модель (Model). Модель предоставляет данные (обычно для View), а также реагирует на запросы (обычно от контроллера), изменяя свое состояние;

- представление (View). Отвечает за отображение информации (пользовательский интерфейс);
- поведение (Controller). Интерпретирует данные, введенные пользователем, и информирует модель и представление о необходимости соответствующей реакции.

Модель данных приложения, логика и представление данных отделены друг от друга.

Одно из ключевых достоинств подобного разделения, что модель не зависит ни от представления, ни от поведения. Это позволяет строить модель независимо от визуального представления, а также создавать несколько различных представлений для одной модели. Формирование web-страниц (View) отделено от выполнения действий по нажатию кнопок на формах (Controller). Общая схема использования паттерна MVC [1] представлена на рис.2.

Взаимодействие «Контроллера» с «Моделью» и с «Видом» может быть асинхронным, а схема использования при переходе от общей к более частным случаям использования в настольных приложениях и далее для web-приложений все более усложняется [1]. Причем MVC-паттерн не дает готового законченного решения, но всегда будет полезен для дисциплины организации разработки.

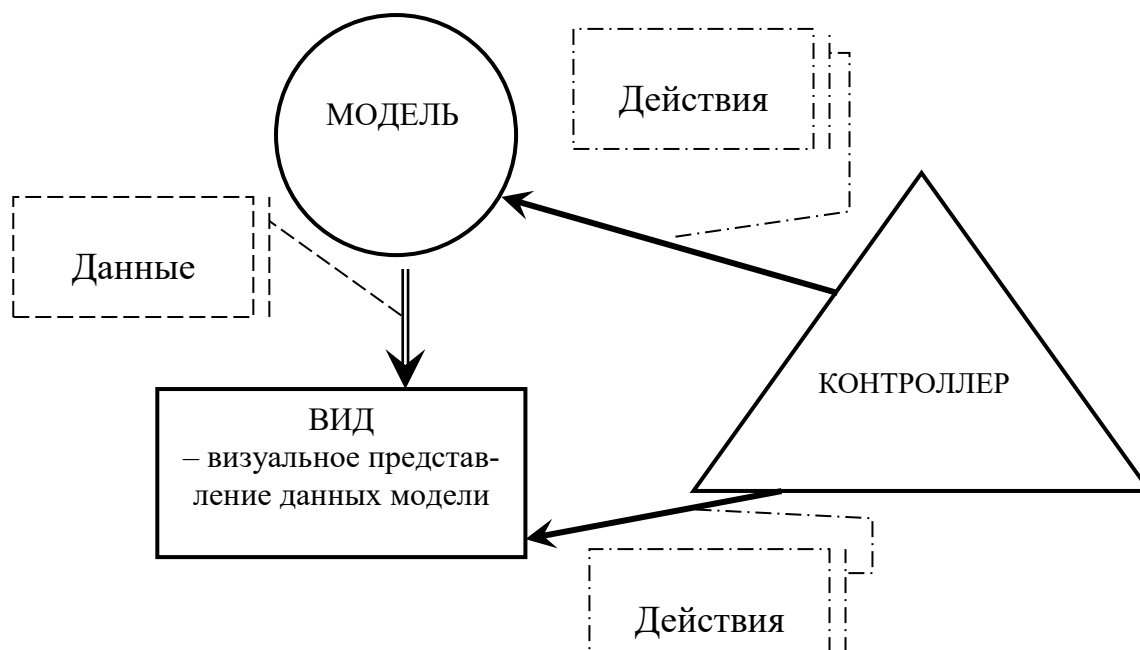


Рис. 2. Общая схема использования паттерна MVC

Библиографический список

1. Овчаренко А.В. AJAX на примерах. – СПб.: БХВ-Петербург, 2009. – 432 с.

И.В. Ганичев, кандидат технических наук, доцент
Е. Г. Троицкая, А. Г. Троицкая, студенты

АНАЛИЗ И СИНТЕЗ ИЕРАРХИЧЕСКОЙ МОДЕЛИ ЭВМ

Введение. В информационных системах (ИС) выделяют подсистему обработки информации, которая состоит из ЭВМ и их комплексов. Анализ концепций построения и функционирования сложных систем неизбежно приводит к иерархическим моделям в форме страт. Известные нам модели страт ЭВМ одномерны, их полюса (верхняя и нижняя страты) отражают отношение «программа – аппаратура», а вот «алгоритмическая» составляющая (важная для представления прикладных программ) вообще не объявлена.

Цели работы – развитие методологии системного анализа применительно к самим стратам и уточнение содержательного описания страт ЭВМ; приспособление полезной модели к нуждам навигации в гипертекстовом описании приложений к учебным курсам на базе страт.

Мы выяснили, что состав и детализация страт зависит от целей их формирования, интересов исследователя, его опыта и квалификации. Так в [3] выделено описание цепочки страт «алгоритм-программа-команда».

В качестве высшей страты, которая соответствует поставленным задачам анализа в рамках последовательных моделей вычислителей, взят абстрактный автомат Тьюринга (общепризнано и другое название – «машина» Тьюринга – МТ) как строго формализованное уточнение понятие алгоритма. Теория алгоритмов и математическая логика образуют фундаментальную основу современной компьютерной базы.

Название «машина» вполне оправдано тем, что алгоритмическое начало, безусловно абстрактного построения Тьюринга соседствует и с «конструктивными» решениями, в которых виден прообраз ЭВМ. Именно конструкция машины Тьюринга (МТ) в абстрактной форме предвосхитила принципы построения современных ЭВМ [3]. В МТ можно выявить сходство подсистем с описанием – «память» в виде ленты с ячейками, в которые определенным образом вводятся символы входного алфавита, «управляющий автомат» с выделенными состояниями, «исполнительное устройство», способное читать ленту и печатать при необходимости новые символы. Для абстрактного автомата составляются программы смены состояний. Программирование абстрактной МТ использует такие основные алгоритмические конструкции как линейная, разветвление и циклы.

Возможно создание систем программирования для имитатора МТ.

Собственно программа и представляет управляющий автомат, но по способу обеспечения наивысшего быстродействия ее лучше хранить в ассоциативной памяти. Таким образом, более подходящей является разнородная память.

В этом намечается первый семантический разрыв абстрактной машины с аналитической моделью – стратой фон Неймана, на уровне которой формулируются принципы построения ЭВМ. Отсюда может быть инициирован учебный процесс синтеза функционально-структурных схем. Вариации формулировки принципов Неймана облегчают поэтапный синтез функциональных структур ЭВМ, соответствующих архитектуре «машины фон Неймана».

Для создания полезной модели-навигатора предварительно преобразуем цепочку страт [1] в форму трехстрочного представления, по сути, «транспонируя» схему данной модели стратификации.

Разработанная стратифицированная модель [1,2] представляет собой обобщенную единую многоуровневую (ОЕМ) модель ЭВМ. Выделение трех ветвей иерархии, соответствующих трем видам обеспечения подсистемы обработки информации ИС, в процедуре своеобразной ортогональной стратификации (появляется дополнительная вертикальная «раскладка» уровней) позволяет определить ОЕМ как трехмерную. Макропредставление ЭВМ на высшей страте ОЕМ показывает потенциальные возможности и философию системы. ОЕМ формирует наглядное представление единства и взаимосвязи всех важных подсистем и компонентов системы.

В системообразующей цепочке страт программного обеспечения (ПО), кроме страты «операционная система», явно показана страта «система программирования», что соответствует принятой классификации ПО и отражает возможность детализации сложившихся представлений о создании прикладного программного обеспечения от «алгоритма» до «команды» путем ее отображения на алгоритмическую ветвь. Для детализации страты «операционная система» необходимо отображение на ресурсы аппаратурных уровней, или создание модели ЭВМ на языке ОС.

Центральное положение страты (рис.1, II ветвь иерархии) операционной системы определено системообразующим значением данной подсистемы. Модель ЭВМ на языке ОС – предмет последующих разработок.

Прикладная программно-алгоритмическая составляющая образует уровни следующей цепочки (см. нижнюю часть рис.). В целом удалось представить все основные виды обеспечения ЭВМ: алгоритмическое – Brainware, программное обеспечение (ПО) – Software, а также аппаратурное – Hardware. Модель включает систему программирования (языки и трансляторы), операционную систему, прикладное ПО.

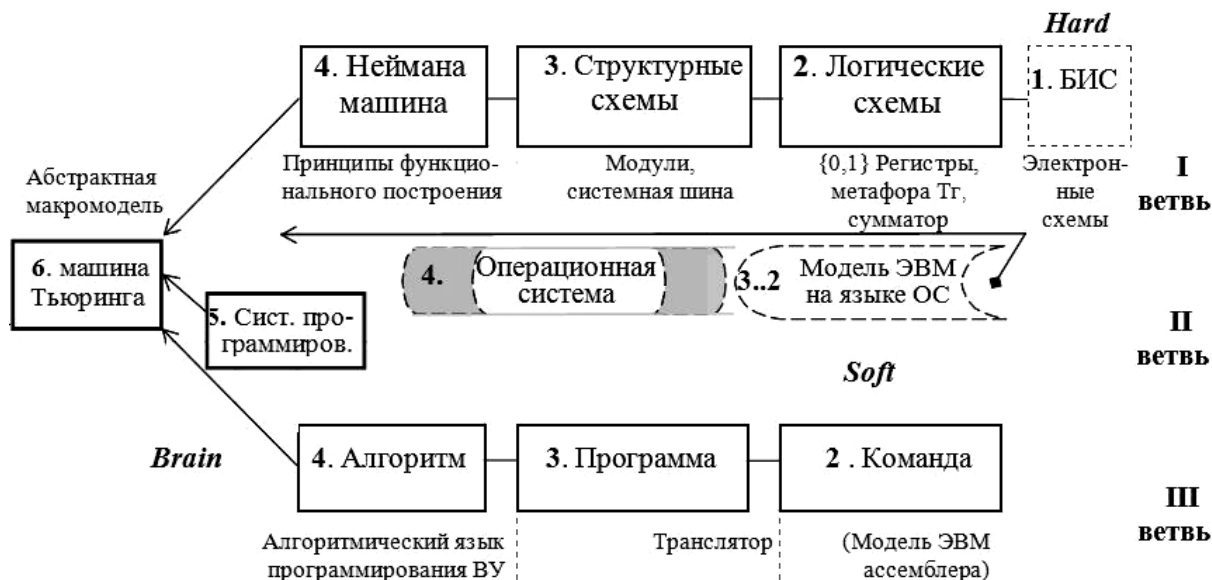


Рис.1. Результаты формирования полезной модели стратификации

Представленная многоуровневая модель ЭВМ позволяет не только показать разновидности обеспечения, но также сохраняет и узловые точки противоречий между концепциями построения подсистем на верхних уровнях и формированием соответствующих компонентов на нижних уровнях иерархии как аппаратного, так и программного обеспечения, поэтому в этом плане модель также является полезной. В результате системного анализа и синтеза структурных и логических схем становятся более ясными особенности архитектуры ЭВМ.

Заключение. Заявленный подход в стратификации моделей наряду с ясным пониманием, в каких областях иерархии зарождается семантический разрыв, позволяет выявить актуальные направления и обратить внимание на развитие архитектуры компьютеров.

Представленная разработка может быть использована как полезная модель для создания на ее основе гипертекста по основным разделам информатики.

Библиографический список

1. Ганичев И.В., Троицкая Е. Г., Троицкая А. Г. Системный анализ иерархической модели ЭВМ. /Информационные системы и технологии: теория и практика: Сборник научных трудов. СПб.: СПбГЛТА, 2008. Стр. 94-102.
2. Аладьев В.З. Хунт Ю.Я., Шишаков М.Л. Основы информатики. Учебное пособие. – М.: Филинь, 1998.
3. Новые подходы к моделированию процессов и корпоративной архитектуры предприятия (Бизнес-инжиниринг и ОРГ-МАСТЕР®: что, зачем, как?) // Григорьев Л. Ю., Кудрявцев Д. В., Спиридонов В. В., Кислова В. В.
http://www.big.spb.ru/publications/bigspb/metodology/new_model_corp_arch.shtml

ПРИМЕНЕНИЕ MICROSOFT SCRIPT EDITOR ДЛЯ РАЗРАБОТКИ И ОТЛАДКИ HTML-СТРАНИЦ

Проблема выбора подходящего инструмента для работы актуальна для многих web-дизайнеров. Разумеется, в принципе для создания полноценного сайта не нужны какие-то специальные программы. Подойдет любой текстовый редактор вроде **Блокнота**. Но работать в Блокноте очень неудобно. Поэтому появилось немало программ специально для создания web-страниц, чтобы время и силы web-дизайнеров в основном тратились на творчество, а не на механическую работу.

Все программы, облегчающие создание **html**-документов, можно условно разделить на три категории.

К первой из них будут относиться программы, которые просто облегчают написание собственно HTML-кода. Они имеют развернутую систему замены и поиска фрагментов текста, могут подсвечивать разными цветами HTML-тэги, отделяя тем самым их от информационного содержания страницы. Таких редакторов очень много, практически все они доступны через Сеть. К сожалению, почти все они являются платными или условно-бесплатными и требуют регистрации, причем стандартные способы продления жизни этих программ в их новых версиях крайне затруднены, - их регистрация происходит через Интернет.

Ко второй категории относятся так называемые **WYSIWYG** – редакторы. В этих редакторах создание web-страницы напоминает создание документа в Microsoft Word: совершенно не нужно думать о тэгах, параметрах. Можно даже не знать, что такое "тэг" и как он выглядит. Страница создается буквально на глазах: достаточно просто написать текст, вставить рисунки или таблицы, оформить строки нужным образом, - и web-страница готова! Но, к сожалению, почти все существующие редакторы типа WYSIWYG генерируют плохой HTML-код – с лишними или даже некорректными тэгами. Кроме того, обеспечить качественное отображение созданных в таких редакторах web-страниц разными браузерами без последующего тщательного редактирования их кода представляет собой практически неразрешимую задачу.

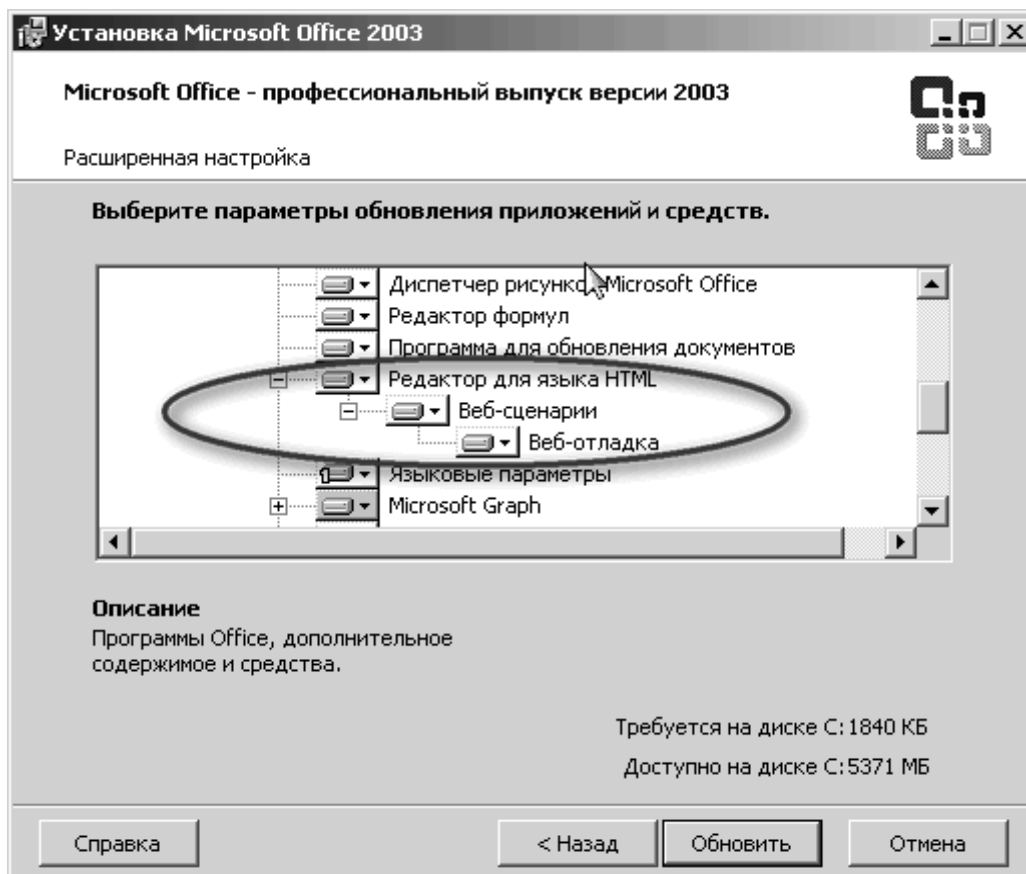
Однако наилучшим выбором для web-дизайнера на настоящий момент являются редакторы, сочетающие в себе возможности как первой, так и второй категории. К ним относится **Microsoft Script Editor** из пакета программ Microsoft Office, начиная с версии 2000.

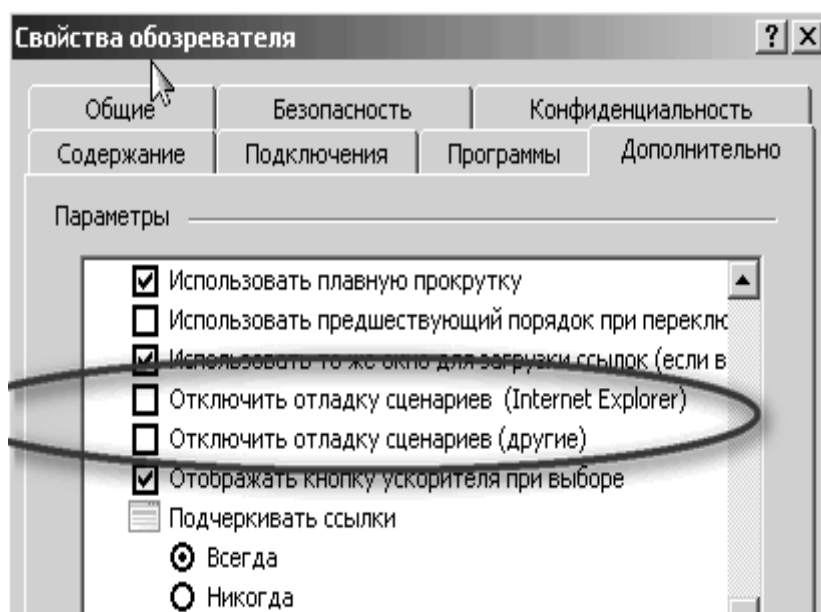
Microsoft Script Editor предназначен для разработки **Web**-сайта,

написания скриптов на языках **JavaScript** и **VBScript** и других Интернет-разработок.

При установке пакета Microsoft Office Script Editor по умолчанию не ставится. Для его установки необходимо выбрать **Выборочную установку**, в меню установки компонентов развернуть пункт **Средства Office**, там – **Редактор HTML-файлов**, а, развернув этот пункт, – **Редактор Web-сценариев**, который пометить как *Запустить с моего компьютера* (см. рис. ниже).

На этом установка Microsoft Script Editor закончена, но необходимо выполнить некоторые настройки **Internet Explorer**. Для этого необходимо запустить его, и в меню **Сервис** выбрать **Свойства обозревателя**. В диалоге настройки открыть вкладку **Дополнительно** и снять две галочки с пунктов: **Отключить отладку сценариев (Internet Explorer)** и **Отключить отладку сценариев (другие)** (см. рис. ниже).





Script Editor дает web-дизайнеру большие возможности. Здесь будут описаны лишь некоторые, самые важные из них.

Если в **Script Editor** создать файл HTML или открыть существующий, то окно с загруженным файлом будет иметь три вкладки. Средняя из них, которая называется **HTML**, показывает HTML-код документа, подсвечивая коричневым цветом тэги, красным - свойства тэгов, а синим - значения свойств (скажем, "font" – это тэг, "size" – это его свойство, а "4" - значение свойства). Левая переводит **Script Editor** в режим **Design** - визуальной разработки web-страницы, а правая - **Quick View** - позволяет оценить будущий вид страницы в браузере.

Окно **HTML** фактически является полноценным редактором кода HTML. Цветовое выделение тэгов весьма удобно для ориентации в тексте. Имеется функция многократной отмены сделанных изменений. На рисунках ниже показаны примеры вида HTML – страницы в окне **HTML** и в окне **Design**.

Miscellaneous Files - Microsoft Script Editor [design] - main00.html

File Edit View Debug Table Tools Window Help

HTML

main00.html

Client Objects & Events (No Events)

```

6 <script language=JavaScript>
7 <!--
8     var a = new Array();      //...?
9     var tn, tk, dt;          //...?
10    var v0n, v0k, dv0;       //...?
11    var q;                   //...?
12    //проверяет корректность введенных данных
13    //если результат отрицательный возвращает false
14
15    function check() {
16        var a_aux = aId.value.split(",");      //...?
17        for(var i = 0; i < a_aux.length; i++) { //...?
18            var tmp = parseFloat(a_aux[i]);    //...?
19            //если не удалось интерпретировать ввод как число,
20            //т.е. tmp содержит Not a Number
21            if( isNaN(tmp) ) { alert("Некорректный ввод"); aId.focus
22            (); return false; } //...?
23            a[i] = tmp; //...?
24        }
25        //аналогично проверим другие поля ввода
26        tn = parseFloat(tId.value) //...?
27        if( isNaN(tn) ) { alert("Некорректный ввод"); tnId.focus();
28        return false; } //...?
29    }
30
31    //...?
32
33    //...?
34
35    //...?
36
37    //...?
38
39    //...?
40
41    //...?
42
43    //...?
44
45    //...?
46
47    //...?
48
49    //...?
50
51    //...?
52
53    //...?
54
55    //...?
56
57    //...?
58
59    //...?
60
61    //...?
62
63    //...?
64
65    //...?
66
67    //...?
68
69    //...?
70
71    //...?
72
73    //...?
74
75    //...?
76
77    //...?
78
79    //...?
80
81    //...?
82
83    //...?
84
85    //...?
86
87    //...?
88
89    //...?
90
91    //...?
92
93    //...?
94
95    //...?
96
97    //...?
98
99    //...?
100
101    //...?
102
103    //...?
104
105    //...?
106
107    //...?
108
109    //...?
110
111    //...?
112
113    //...?
114
115    //...?
116
117    //...?
118
119    //...?
120
121    //...?
122
123    //...?
124
125    //...?
126
127    //...?
128
129    //...?
130
131    //...?
132
133    //...?
134
135    //...?
136
137    //...?
138
139    //...?
140
141    //...?
142
143    //...?
144
145    //...?
146
147    //...?
148
149    //...?
150
151    //...?
152
153    //...?
154
155    //...?
156
157    //...?
158
159    //...?
160
161    //...?
162
163    //...?
164
165    //...?
166
167    //...?
168
169    //...?
170
171    //...?
172
173    //...?
174
175    //...?
176
177    //...?
178
179    //...?
180
181    //...?
182
183    //...?
184
185    //...?
186
187    //...?
188
189    //...?
190
191    //...?
192
193    //...?
194
195    //...?
196
197    //...?
198
199    //...?
200
201    //...?
202
203    //...?
204
205    //...?
206
207    //...?
208
209    //...?
210
211    //...?
212
213    //...?
214
215    //...?
216
217    //...?
218
219    //...?
220
221    //...?
222
223    //...?
224
225    //...?
226
227    //...?
228
229    //...?
230
231    //...?
232
233    //...?
234
235    //...?
236
237    //...?
238
239    //...?
240
241    //...?
242
243    //...?
244
245    //...?
246
247    //...?
248
249    //...?
250
251    //...?
252
253    //...?
254
255    //...?
256
257    //...?
258
259    //...?
260
261    //...?
262
263    //...?
264
265    //...?
266
267    //...?
268
269    //...?
270
271    //...?
272
273    //...?
274
275    //...?
276
277    //...?
278
279    //...?
280
281    //...?
282
283    //...?
284
285    //...?
286
287    //...?
288
289    //...?
290
291    //...?
292
293    //...?
294
295    //...?
296
297    //...?
298
299    //...?
300
301    //...?
302
303    //...?
304
305    //...?
306
307    //...?
308
309    //...?
310
311    //...?
312
313    //...?
314
315    //...?
316
317    //...?
318
319    //...?
320
321    //...?
322
323    //...?
324
325    //...?
326
327    //...?
328
329    //...?
330
331    //...?
332
333    //...?
334
335    //...?
336
337    //...?
338
339    //...?
340
341    //...?
342
343    //...?
344
345    //...?
346
347    //...?
348
349    //...?
350
351    //...?
352
353    //...?
354
355    //...?
356
357    //...?
358
359    //...?
360
361    //...?
362
363    //...?
364
365    //...?
366
367    //...?
368
369    //...?
370
371    //...?
372
373    //...?
374
375    //...?
376
377    //...?
378
379    //...?
380
381    //...?
382
383    //...?
384
385    //...?
386
387    //...?
388
389    //...?
390
391    //...?
392
393    //...?
394
395    //...?
396
397    //...?
398
399    //...?
400
401    //...?
402
403    //...?
404
405    //...?
406
407    //...?
408
409    //...?
410
411    //...?
412
413    //...?
414
415    //...?
416
417    //...?
418
419    //...?
420
421    //...?
422
423    //...?
424
425    //...?
426
427    //...?
428
429    //...?
430
431    //...?
432
433    //...?
434
435    //...?
436
437    //...?
438
439    //...?
440
441    //...?
442
443    //...?
444
445    //...?
446
447    //...?
448
449    //...?
450
451    //...?
452
453    //...?
454
455    //...?
456
457    //...?
458
459    //...?
460
461    //...?
462
463    //...?
464
465    //...?
466
467    //...?
468
469    //...?
470
471    //...?
472
473    //...?
474
475    //...?
476
477    //...?
478
479    //...?
480
481    //...?
482
483    //...?
484
485    //...?
486
487    //...?
488
489    //...?
490
491    //...?
492
493    //...?
494
495    //...?
496
497    //...?
498
499    //...?
500
501    //...?
502
503    //...?
504
505    //...?
506
507    //...?
508
509    //...?
510
511    //...?
512
513    //...?
514
515    //...?
516
517    //...?
518
519    //...?
520
521    //...?
522
523    //...?
524
525    //...?
526
527    //...?
528
529    //...?
530
531    //...?
532
533    //...?
534
535    //...?
536
537    //...?
538
539    //...?
540
541    //...?
542
543    //...?
544
545    //...?
546
547    //...?
548
549    //...?
550
551    //...?
552
553    //...?
554
555    //...?
556
557    //...?
558
559    //...?
560
561    //...?
562
563    //...?
564
565    //...?
566
567    //...?
568
569    //...?
570
571    //...?
572
573    //...?
574
575    //...?
576
577    //...?
578
579    //...?
580
581    //...?
582
583    //...?
584
585    //...?
586
587    //...?
588
589    //...?
590
591    //...?
592
593    //...?
594
595    //...?
596
597    //...?
598
599    //...?
600
601    //...?
602
603    //...?
604
605    //...?
606
607    //...?
608
609    //...?
610
611    //...?
612
613    //...?
614
615    //...?
616
617    //...?
618
619    //...?
620
621    //...?
622
623    //...?
624
625    //...?
626
627    //...?
628
629    //...?
630
631    //...?
632
633    //...?
634
635    //...?
636
637    //...?
638
639    //...?
640
641    //...?
642
643    //...?
644
645    //...?
646
647    //...?
648
649    //...?
650
651    //...?
652
653    //...?
654
655    //...?
656
657    //...?
658
659    //...?
660
661    //...?
662
663    //...?
664
665    //...?
666
667    //...?
668
669    //...?
670
671    //...?
672
673    //...?
674
675    //...?
676
677    //...?
678
679    //...?
680
681    //...?
682
683    //...?
684
685    //...?
686
687    //...?
688
689    //...?
690
691    //...?
692
693    //...?
694
695    //...?
696
697    //...?
698
699    //...?
700
701    //...?
702
703    //...?
704
705    //...?
706
707    //...?
708
709    //...?
710
711    //...?
712
713    //...?
714
715    //...?
716
717    //...?
718
719    //...?
720
721    //...?
722
723    //...?
724
725    //...?
726
727    //...?
728
729    //...?
730
731    //...?
732
733    //...?
734
735    //...?
736
737    //...?
738
739    //...?
740
741    //...?
742
743    //...?
744
745    //...?
746
747    //...?
748
749    //...?
750
751    //...?
752
753    //...?
754
755    //...?
756
757    //...?
758
759    //...?
760
761    //...?
762
763    //...?
764
765    //...?
766
767    //...?
768
769    //...?
770
771    //...?
772
773    //...?
774
775    //...?
776
777    //...?
778
779    //...?
780
781    //...?
782
783    //...?
784
785    //...?
786
787    //...?
788
789    //...?
790
791    //...?
792
793    //...?
794
795    //...?
796
797    //...?
798
799    //...?
800
801    //...?
802
803    //...?
804
805    //...?
806
807    //...?
808
809    //...?
810
811    //...?
812
813    //...?
814
815    //...?
816
817    //...?
818
819    //...?
820
821    //...?
822
823    //...?
824
825    //...?
826
827    //...?
828
829    //...?
830
831    //...?
832
833    //...?
834
835    //...?
836
837    //...?
838
839    //...?
840
841    //...?
842
843    //...?
844
845    //...?
846
847    //...?
848
849    //...?
850
851    //...?
852
853    //...?
854
855    //...?
856
857    //...?
858
859    //...?
860
861    //...?
862
863    //...?
864
865    //...?
866
867    //...?
868
869    //...?
870
871    //...?
872
873    //...?
874
875    //...?
876
877    //...?
878
879    //...?
880
881    //...?
882
883    //...?
884
885    //...?
886
887    //...?
888
889    //...?
890
891    //...?
892
893    //...?
894
895    //...?
896
897    //...?
898
899    //...?
900
901    //...?
902
903    //...?
904
905    //...?
906
907    //...?
908
909    //...?
910
911    //...?
912
913    //...?
914
915    //...?
916
917    //...?
918
919    //...?
920
921    //...?
922
923    //...?
924
925    //...?
926
927    //...?
928
929    //...?
930
931    //...?
932
933    //...?
934
935    //...?
936
937    //...?
938
939    //...?
940
941    //...?
942
943    //...?
944
945    //...?
946
947    //...?
948
949    //...?
950
951    //...?
952
953    //...?
954
955    //...?
956
957    //...?
958
959    //...?
960
961    //...?
962
963    //...?
964
965    //...?
966
967    //...?
968
969    //...?
970
971    //...?
972
973    //...?
974
975    //...?
976
977    //...?
978
979    //...?
980
981    //...?
982
983    //...?
984
985    //...?
986
987    //...?
988
989    //...?
990
991    //...?
992
993    //...?
994
995    //...?
996
997    //...?
998
999    //...?
1000

```

Design HTML

Ready Ln 22 Col 35 Ch 26 INS

Miscellaneous Files - Microsoft Script Editor [design] - main00.html

File Edit View Debug Format Table Insert Tools Window Help

HTML

main00.html

Зависимость пройденного расстояния S при равноускоренном движении от времени t , ускорения a и начальной скорости v_0

Ускорение a
 [м/сек²]:
 (в этом поле через запятую перечислите различные значения ускорения)

Время t [сек] изменять
 от начального значения t_n : до конечного значения t_k : с
 шагом Δt :

Начальную скорость v_0 [м/сек] изменять
 от начального значения v_{0n} : до конечного значения v_{0k} :
 с шагом Δv_0 :

div

Design HTML

Ready

Для того, чтобы добавить в текст страницы код, соответствующий какому-нибудь элементу, достаточно всего лишь дважды кликнуть мышью на название этого элемента на панели **Toolbox**. Эта панель по умолчанию располагается слева окна **Script Editor**.

Если в стандартном **HTML** не окажется средств для задания какому-либо элементу нужных свойств, то будет использован язык каскадных таблиц стилей – **CSS**. Тогда при выборе из контекстного меню пункта **Properties** отобразится не окно задания свойств тэга, а большая панель свойств **Properties**, где можно задать огромное количество параметров тэга.

Превосходные средства работы с Html-кодом - это лишь очень малая часть возможностей **Script Editor** (хотя и их вполне бы хватило на небольшой web-редактор, причем очень даже неплохой для своего класса)! Вкладка **Design** основного окна этой программы представляет собой полноценный WYSIWYG-редактор Web-страницы, многократно превосходящий по своим возможностям большинство других аналогов.

Особую уникальность **Script Editor** придает то, что помимо web-редактора он представляет из себя еще и среду программирования, по набору функций похожую на среду **Visual Basic for Applications**. Рассчитана эта среда на два языка – **JavaScript** и **VBScript**.

Те, кто уже писал макросы для Microsoft Word, найдут в **Script Editor** много общего с встроенным в **Word** редактором **Visual Basic for Applications**. И это понятно – Microsoft стремится создать единую среду программирования для разработки приложений на любых языках, включая в новые редакции своих продуктов лучшее из предыдущих версий. Кроме того, web-сценарии фактически представляют собой "макросы для браузера", так как они не компилируются заранее, а интерпретируются браузером после загрузки web-страницы.

Чтобы добавить на страницу сценарий, даже не надо самому писать тэги. Можно просто кликнуть правой кнопкой мыши на нужном месте и из контекстного меню выбрать **Script Block-Client** (второй вариант этого меню, **Server**, относится к возможности реализации технологии ASP). Для более легкой ориентировки на странице, где имеется множество скриптов, очень хорошо использовать окно **Script Outline** (вызывается из меню **View-Other Window-Script Outline**), в котором перечислены все скрипты, находящиеся в документе. Чтобы перейти к нужному, достаточно пары кликов мыши, при этом курсор в окне HTML установится на его начало.

(Технология **Dynamic Html** неразрывно связана с технологиями **JavaScript** и **VBScript**. Иными словами, можно сказать, что **Dynamic Html** позволяет обратиться к любому объекту на странице, а **VBScript** и **JavaScript** - что-нибудь с этим объектом сделать. Поэтому сценарии на **VBScript** и **JavaScript** часто включают в себя элементы **Dynamic Html** - обращения к поименованным элементам страницы.)

При написании самого текста сценария **Script Editor** как бы "ведет" программиста по программе, подсказывая ему возможные продолжения команд. Подобная система используется в большинстве современных средств разработки приложений, так что те, кто ранее программировал на других языках, окажутся в своей привычной среде, а те, кто ранее не занимался созданием программ, получают возможность познакомиться с настоящей средой программирования.

Сценарий - это самая настоящая программа, и, к сожалению, при ее написании ошибки практически неизбежны. В **Script Editor** есть мощная система отладки сценариев, мало чем уступающая аналогичным системам других сред программирования. Для того чтобы установить местонахождение ошибки, используются "точки останова" и контрольные значения, а чтобы разобраться в сложной странице, где есть несколько скриптов, вызывающих один другой - окно стека вызова – **Call Stack**. В окне редактора можно пометить с помощью мыши на сером поле у левого края окна определенные строки сценария (поставить "точку останова"), и в этом случае при загрузке страницы с этим сценарием в браузер из среды **Script Editor** (это можно сделать, нажав клавишу **F5**) в момент, когда выполнение сценария дойдет до отмеченной строки, его работа остановится и активируется окно редактора кода. Задержав в этот момент курсор мыши над какой-нибудь переменной из кода сценария, во всплывающей подсказке можно увидеть ее значение. Также значение любой переменной показывается в окне контрольных значений **Locals**.

Выполнять сценарий по строчкам, чтобы поточнее определить место ошибки, труда не составит - богатые возможности меню **Debug** к вашим услугам.

Всех перечисленных возможностей **Script Editor** с лихвой хватит подавляющему большинству разработчиков web-страниц

Для чисто практического использования с целью создания web-страниц **Script Editor** подходит просто идеально. Тем более что этот продукт не надо искать по всем рынкам и лоткам, спрашивая у всех компакт-диск с ним - он входит в стандартную поставку **Microsoft Office** и таким образом имеется на любом компакт-диске с этим пакетом программ. Просто установите **Microsoft Office** и вы получите также и **Script Editor**, если укажете необходимость его установки в диалоговом окне инсталлятора.

В заключении можно сказать, что **Script Editor** сочетает в себе лучшие свойства редакторов web-страниц, как текстовых, так и типа WYSIWYG, а также полноценной среды программирования web-сценариев, что делает этот инструмент практически универсальным. Кроме того, он входит в стандартную поставку пакета **Microsoft Office**, начиная с версии 2000, а значит, есть у каждого, у кого установлен этот пакет.

ИНФОРМАЦИОННЫЙ КИОСК КАФЕДРЫ ИИС ЛТА

Обеспечение высокого качества образовательных услуг – основная цель любого учебного заведения. Тем более это относится к высшим учебным заведениям, где необходимо применять самые передовые и эффективные технологии, используя последние достижения науки и техники. Одной из таких новинок в мире информации являются информационные киоски.

Информационные киоски стали неотъемлемой частью жизни современного общества и широко распространены во всем мире. Они служат для выполнения разнообразных функций электронных помощников человека.

Информационный киоск – это многофункциональное и многоцелевое электронное устройство, имеющее антивандальный корпус, в напольном или настенном исполнении. В зависимости от дизайна и функционального назначения они могут содержать то или иное оборудование, однако, основной отличительной чертой их являются сенсорные экраны (Touch screen), позволяющие простым прикосновением запрашивать интересующую информацию.

При отсутствии или невозможности использования сенсорного экрана устройством ввода (запроса) для информационного киоска является стандартная клавиатура или мышь в антивандальном исполнении.

Обычной и наиболее характерной задачей информационного киоска является предоставление информации.

В учебных заведениях одним из перспективных направлений использования информационного киоска является визуализация *электронного расписания занятий* с сервисом и опциями, не доступными его классическому настенному воплощению, например, такими как применение фильтров, связь с кафедральным сервером и сервером ЛТА, а также выход в Internet.

Разрабатываемый на кафедре ИИС информационный киоск предполагает наличие следующих сервисных возможностей:

- демонстрацию расписания на текущий день
- вызов расписания на любой запрашиваемый день недели
- использование фильтра для вызова расписания занятий конкретной учебной группы по дням недели
- использование фильтра для вызова расписания работы любого преподавателя по дням недели
- просмотр свободных аудиторий (классов) на текущий или любой другой день

- оперативное внесение изменений в расписание в случае болезни преподавателя или других непредвиденных случаях
- расписание консультаций, зачетов и экзаменов
- возможность корректировки расписания методистом кафедры, заведующим кафедрой или любым преподавателем (при согласовании с зав. кафедрой) на своем рабочем месте при наличии компьютера, подключенного к локальной сети.

Составление учебного расписания является весьма сложной математической задачей. Вследствие этого, как правило, на практике использовалась и используется технология составления расписания вручную, которая предполагает сбор и классификацию исходной информации, составление вспомогательных таблиц, непосредственно составление расписания, его проверку и корректировку. При такой технологии очень трудно получить расписание, которое отражало бы все особенности данного учебного заведения и в полной мере учитывало пожелания преподавателей и потребности студентов.

В последние годы на рынке программ для образовательных учреждений стали появляться системы автоматизированного формирования расписаний, которые поставляются как самостоятельные программные продукты или как элементы систем управления учебным учреждением.

Недостатком первого подхода является необходимость ручного ввода данных, в то время как во втором случае эти сведения уже содержатся в базе данных.

Технология составления расписания занятий с использованием компьютера в общем случае может быть представлена следующим образом:

1. Ввод исходных данных.
2. Ручной ввод отдельных элементов расписания (при необходимости).
3. Автоматическое составление расписания.
4. Контроль качества полученного расписания.
5. Внесение изменений в исходные данные и повтор п.п. 1-4.
6. Ручная корректировка расписания (при необходимости).

Оставляя за рамками данной статьи математический подход к существующей проблеме, в настоящей разработке предлагается применить полуавтоматический метод составления расписания на базе широко распространенного табличного процессора Excel, что значительно упрощает работу методиста кафедры по созданию первичного расписания занятий, а также его корректировке.

Суть его заключается в следующем.

Используя полученные от диспетчера исходные данные, методист вносит элементы расписания в заранее подготовленные шаблоны, разработанные сотрудниками кафедры.

Поскольку в ЛТА расписание подразделяется на чётные и нечётные недели, таких шаблонов может быть от 10 (пятидневная неделя) до 12 (шестидневная неделя).

На этом работа методиста по составлению расписания заканчивается. Всё остальное заложено в программном обеспечении компьютера. Это набор функций, реализованных в виде макросов Excel, и гиперссылки, позволяющие продемонстрировать электронное расписание и его производные на экране монитора или при необходимости выводить на печать.

На рис. 1 представлен вид экрана информационного киоска в режиме демонстрации расписания на текущий день.

Сегодня		22.12.2008				Нечётная неделя							
Понедельник													
		9.15-10.50		11.10-12.45		13.30-15.05		15.15-16.50		17.00-18.35		18.45-20.20	
Дисциплина		Физика										Базы данных	
Преподаватель		Иванов										Сидоров	
Аудитория		231										7	
Студенты		ФЗУ 1.4										ХТФ 5-2	
Дисциплина		Введение в ВО		Базы данных									
Преподаватель		Петров		Сидоров									
Аудитория		3		7									
Студенты		ЛТД 3-5		ХТФ 5-2									
Дисциплина													
Преподаватель													
Аудитория										Физика			
Студенты										Иванов			
Дисциплина										231			
Преподаватель										ФЗУ 1.4			
Аудитория													
Студенты													
Дисциплина								Информатика					
Преподаватель								Заяц А.М.					
Аудитория								6					
Студенты								ФЗУ 1-2					
Сегодня		Любой день	Изменить	Аудитории свободны	Преподаватель	Зачеты	Экзамен	Задолженность					

Рис. 1

Как видно из рисунка, на экране отображено полноценное и подробное расписание занятий на текущий день со всеми его атрибутами, расписание, исключая влияние человеческого фактора (функция <СЕГОДНЯ> не ошибается!).

Для удобства восприятия отдельные элементы расписания могут быть выделены цветом.

В нижней части экрана размещено пиктографическое меню, каждая кнопка которого представляет собой гиперссылку на опцию, соответствующую её названию.

В описываемом случае была нажата кнопка «Сегодня» (Затемнена).

При нажатии кнопки «Любой день» открывается контекстное меню (Пн., Вт., Ср., Чт., Пт., Сб.), позволяющее просмотреть расписание на любой заданный день чётной (два нажатия) или нечётной (одно нажатие) недели. Картинка будет аналогичной, с указанием вызванного дня недели. Отсутствует, разумеется, только подсказка «СЕГОДНЯ».

Кнопка «Преподаватель» позволяет ознакомиться с расписанием работы каждого преподавателя на чётную (два нажатия) или нечётную (одно нажатие) неделю. (Рис.2).

Преподаватель	Понедельник		Вторник		Среда		Четверг		Пятница		Суббота	
	Время	Ауд.	Время	Ауд.	Время	Ауд.	Время	Ауд.	Время	Ауд.	Время	Ауд.
Петров	9.15-10.50//	231	13.30-15.05//	6			11.10-12.45//	3				
Сидоров	17.00-18.35//	2										
Синицин					9.15-10.50//	231			15.15-16.50//	231		
Морозов	11.10-12.45//	145										
Карасев							13.30-15.05//	8			11.10-12.45//	45
Иванов			13.30-15.05//	50								
*												
*												
*												
*												
*												
*	*		*		*		*		*		*	
*												
*												
*												
Иевлев			9.15-10.50//	231			17.00-18.35//	56				
Константинов	15.15-16.50//	7										
Правдин					11.10-12.45//	2						
ч ё т н я я н е д е л я												
Сегодня	Любой день	Изменить	Аудитории свободны	Преподаватель	Зачеты	Экзамен	Задолженность					

Рис. 2

Для проведения дополнительных занятий, консультаций или любых других незапланированных мероприятий может понадобиться свободная аудитория или класс (из числа эксплуатируемых кафедрой ИИС).

Для этой цели служит кнопка «Аудитории свободны».

При нажатии этой кнопки раскрывается контекстное меню, аналогичное опции «Любой день», позволяющее просмотреть свободные помещения на любой заданный день чётной или нечётной недели (Рис.3.)

Кнопка «Изменить» недоступна. Она используется только методом кафедры или зав.кафедрой со своего рабочего места. В исключительных случаях она может быть использована преподавателем при согласо-

нии с заведующим кафедрой для мобильной корректировки расписания в случае непредвиденных обстоятельств.

Кнопки «Зачёты» и «Экзамены» доступны пользователям информационного киоска только для просмотра. Управление этими кнопками является прерогативой методиста кафедры.

Из других функций киоска можно отметить возможность ознакомления студентов с историей и традициями кафедры, с литературой и учебными пособиями, необходимыми для освоения той или иной дисциплины, оперативное оповещение приказов и распоряжений ректората, деканата и т.п.

	Понедельник		Вторник		Среда		Четверг		Пятница		Суббота		
	Время	Ауд.	Время	Ауд.	Время	Ауд.	Время	Ауд.	Время	Ауд.	Время	Ауд.	
С В О Б О Д Н Ы													
		17.00-18.35// 2											
										15.15-16.50// 231			
								13.30-15.05// 8				11.10-12.45// 45	
			13.30-15.05// 50										
		9.15-10.50// 231											
										11.10-12.45// 145			
			9.15-10.50// 231										
					9.15-10.50// 231								
												13.30-15.05// 6	
								17.00-18.35// 56					
		15.15-16.50// 7											
					11.10-12.45// 2								
	чётная неделя												
Сегодня	Любой день	Изменить	Аудитории свободны	Преподаватель	Зачеты	Экзамен	Задолженность						

Рис. 3

Конструктивно информационный киоск кафедры ИИС предполагается выполнить как выносной терминал, в качестве которого может использоваться обычный жидкокристаллический дисплей либо плазменная панель.

Выносной терминал соединяется с системным блоком, подключённым к локальной сети и управляемым с рабочего места методиста кафедры, а при необходимости, как уже упоминалось, с рабочего места любого заинтересованного лица, имеющего на то право и знающего пароль администратора.

В качестве устройства ввода предполагается использовать стандартную клавиатуру или мышь в антивандальном исполнении.

Один из вариантов конструктивного выполнения информационного киоска кафедры ИИС ЛТА представлен на рис. 4.



Рис. 4

В перспективе предполагается подключение процессора информационного киоска к локальной сети Академии с целью оперативного обмена информацией и возможности выхода в Internet. (Рис.5)

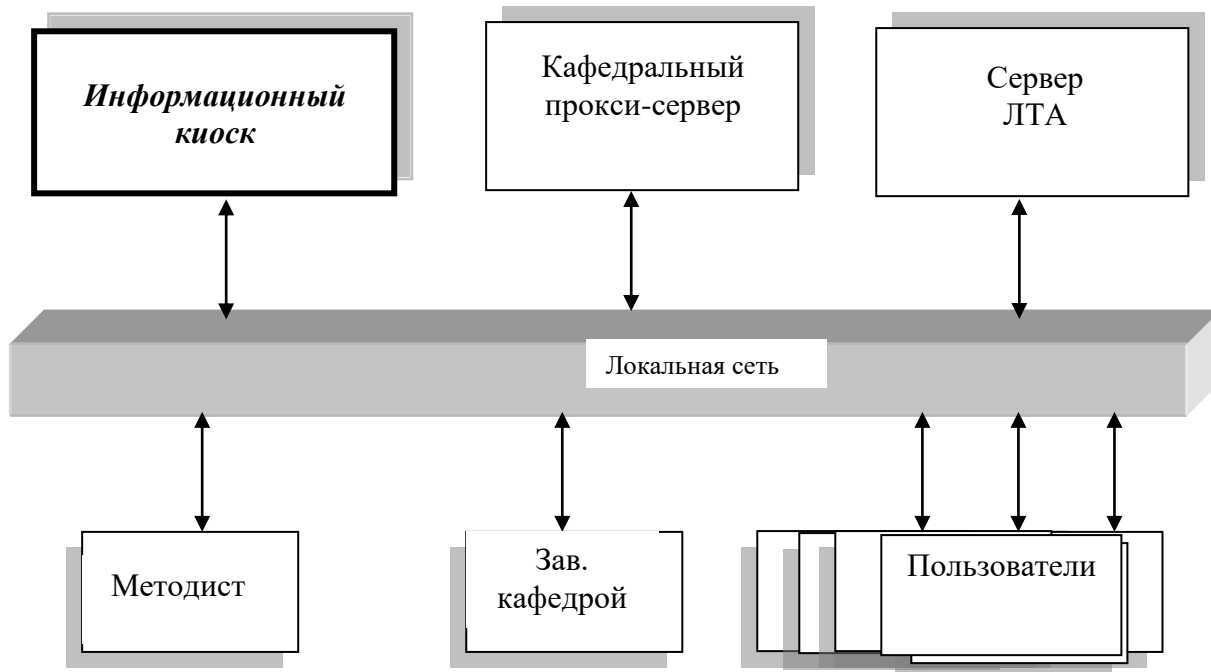


Рис. 5

Локальная сеть обеспечивает связь информационного киоска с заведующим кафедрой, с методистом кафедры, с зарегистрированными пользователями, с кафедральным прокси-сервером, а также с сервером ЛТА, что, в свою очередь, предоставляет возможность выхода в Internet.

Использование электронного расписания имеет следующие преимущества:

1. Уменьшение трудозатрат на разработку
2. Удобство восприятия
3. Широкий набор вспомогательных функций
4. Исключение человеческого фактора в решении вопроса – чётная или нечётная неделя
5. Возможность дистанционного внесения изменений

Курилова М.Н., старший преподаватель

ЦЕЛИ, ЗАДАЧИ И МЕТОДИКА ПРОВЕДЕНИЯ ОЛИМПИАД ПО ИНФОРМАТИКЕ

Изучение информатики и информационных технологий в настоящее время является обязательным условием качественной подготовки специалистов. Целевая установка вузовских стандартов – изучение студентами современного состояния и эффективного использования информационных технологий для решения будущих профессиональных задач. В отраслевом вузе, каковым является Санкт-Петербургская лесотехническая академия, информатика – общеобразовательный предмет и студент не всегда уделяет должное внимание изучению этой дисциплины.

Кафедра информатики и информационных систем СПбГЛТА последние годы серьезно занимается подготовкой, организацией и проведением олимпиад по информатике для студентов всех факультетов и специальностей.

Проведение олимпиад по информатике позволяет значительно повысить интерес к предмету, но в условиях СПбГЛТА имеет свою специфику. Традиционные олимпиады по информатике, как правило, ориентированы на поиск программистов, то есть студентов, знающих языки программирования, имеющих неплохие познания в математике и умеющих строить оптимальные алгоритмы при решении всевозможных задач. Такие олимпиады хороши при подготовке будущих профессионалов в области информационных технологий, в них обычно принимают участие студенты, которые со школьных лет активно интересуются программированием и IT-техноло-

гиями. В наших же условиях, когда знания студента бывают весьма скромными, и, к сожалению, часто недотягивают даже до уровня обычной школьной общеобразовательной программы, необходимо, прежде всего, заинтересовать его предметом.

За годы проведения соревнований удалось сформулировать основные **цели и задачи олимпиад** в том случае, когда информатика является не профилирующим, а только общеобразовательным предметом.

Основной задачей таких олимпиад является повышение общего уровня подготовки студентов по информатике и информационным технологиям.

Конечно, не нужно забывать о цели любых олимпиад – выявление наиболее одаренных, подготовленных, интересующихся предметом студентов. В дальнейшем наиболее талантливые и заинтересованные студенты могут быть привлечены к научной работе.

Олимпиада должна способствовать развитию навыков по самостоятельному приобретению знаний (самообучению). Необходимо стимулировать более углубленное изучение предмета, подсказывать, какие разделы пропущены и требуют самостоятельной доработки. Это учитывается при составлении заданий и в правилах проведения олимпиады: студентам, набравшим по темам курса информатики балл не ниже указанного, выставляется оценка за экзамен по результатам олимпиады, и они освобождаются от сдачи экзамена. При этом организуется подготовка к олимпиаде, студенты могут получить подробный перечень вопросов, которые будут рассматриваться в заданиях, а при необходимости консультацию преподавателя. Таким образом, студент занявший, например, десятое место может быть аттестован по курсу информатики с оценкой хорошо или отлично.

Такой подход позволяет усилить связь между информатикой и другими, смежными, дисциплинами, например, при подготовке к олимпиаде выясняется, что для решения задачи по информатике требуются знания по какому-либо изученному ранее разделу высшей математики или физики, а их нет, приходится ликвидировать и этот пробел в образовании.

Олимпиада должна способствовать развитию интереса к будущей профессиональной деятельности, это требование нужно учитывать при подборе олимпиадных задач.

Участники олимпиады получают практический опыт работы в условиях соревнований, когда нужно грамотно распределить время и силы, быстро принимать правильные решения. Такие навыки полезны в любой профессиональной сфере в условиях конкуренции и нехватки времени.

Проведение олимпиад дает представление о подготовке, полученной студентами в школе, а также позволяет проверить текущий уровень преподавания информатики: это реальный способ оценить качество читаемых курсов и программ, положенных в их основу.

Ежегодное проведение олимпиад способствует отработке конкурсных заданий, дает возможность сформировать банк заданий (решений). Полученный банк можно использовать при проведении теоретических и практических занятий, для проведения текущей и итоговой аттестации.

Чтобы максимально реализовать цели и задачи подобных соревнований, важен четкий и продуманный **порядок проведения олимпиады**, в этом плане кафедрой также наработан некоторый опыт.

Олимпиада должна состоять из двух туров – теоретического и практического, и студент, по желанию, может участвовать в одном туре или обоих. Такой порядок позволяет выявить студентов, умеющих решать практические задачи, владеющих различными приложениями Windows, но не имеющих достаточных теоретических знаний. Как показывает опыт проведения олимпиад, студент, выбравший теоретический тур, часто готов и к практической части, а вот наоборот, реже.

Оба тура олимпиады проводятся в один день, обычно во второй половине дня, на каждый вид соревнований отводится два часа, начинается олимпиада с теории. Но при этом студент, явившийся только на практический тур, может сразу приступить к выполнению заданий.

Теоретическая информатика проводится в режиме письменного экзамена, что позволяет оценить не только результаты участника, но и его умение мыслить и выбирать оптимальное решения. Правда, такой режим проведения олимпиады, а не тестовый, доставляет больше хлопот жюри.

Для практической информатики каждому участнику выделяется компьютер, на котором он должен за отведенное время выполнить максимальное количество заданий.

На каждом туре студент может выполнять задания в произвольном порядке. Такой подход даже рекомендуется, ведь в условиях ограниченного времени, желательно сделать как можно больше заданий. При оценке результатов становится понятно, какие задания вызывают больше затруднений.

Во время олимпиады категорически запрещается пользоваться личными компьютерами, калькуляторами, электронными записными книжками, средствами связи (пейджерами, мобильными телефонами и т.п.). При составлении заданий это учитывается – сложных математических расчетов в заданиях не требуется, а консультанты за пределами аудитории не приветствуются – каждый участник должен продемонстрировать собственные знания. Нарушители этих правил удаляются и из числа участников исключаются.

На практическом туре необходимо предусмотреть возможные сбои в работе компьютера или используемого программного обеспечения, возникающие не по вине участника. Устранение неполадок требует времени, по-

этому по решению жюри, в таких случаях, участнику компенсируется сбойное время.

Подбор и формирование **олимпиадных заданий** работа трудоемкая и кропотливая, удачные, хорошо сформулированные задания позволяют сделать олимпиаду интересной и познавательной, а также правильно оценить уровень подготовки участников.

Прежде всего, требуется четкая и понятная формулировка заданий, исключающая неоднозначное понимание их условий. В каждом задании должна быть хорошо понятна основная мысль, все второстепенное, менее значимое не должно отвлекать от понимания и решения. Неплохо, если задания могут быть решены несколькими способами (многовариантность решения), среди которых есть оптимальные. При оценке результатов естественно выше оценивается оптимальное решение.

Постановку задач желательно выполнить в увлекательной, интересной форме, их идеи должны быть по возможности оригинальны, не только формулировкой, но и решением, хотя эти пожелания могут представлять трудность для составителей – они требуют от них высокой квалификации и изобретательности.

В тексте заданий не должны встречаться термины и понятия, выходящие за пределы базового курса изучаемых предметов, в крайнем случае, они должны быть четко определены и конкретизированы.

Необходимо наличие заданий различного уровня сложности. Например, требуются одна-две задачи доступные каждому участнику («утешительные задачи»). Кроме этого несколько задач должны быть достаточно сложными – чтобы не огорчить тех, кто пришел за интересными заданиями, и хотел бы блеснуть знаниями. В общем, участникам должен предлагаться набор добротных заданий на различные темы, чтобы каждый мог найти интересные и доступные для себя задания, а жюри понять, какие темы курсы хорошо изучены, а какие требуют доработки.

Предложенные задания должны охватывать все основные темы курса информатики: информация и сообщение, кодирование и декодирование информации, системы счисления, основы алгебры логики, архитектура ЭВМ, алгоритмизация, языки программирования и методы трансляции, программное обеспечение и программирование, модели, моделирование, компьютерные сети и т.д. Желательно, чтобы в отдельных заданиях использовались сразу несколько тем не только курса информатики, но и смежных дисциплин, например, математики, физики и т. д. Олимпиады, которые не учитывают этого требования, рискуют «получить» в качестве победителя узкого специалиста, который имеет серьезные пробелы в знаниях.

Неплохо иметь одно задание, поддерживающее весь цикл решения задачи по информатике – от содержательной постановки до получения и

анализа решения. В постановке задачи желательно ограничение операционной среды для решения до минимально необходимого (ограничение ресурсов и средств), что позволит выявить уровень теоретических знаний, умение практически использовать ресурсы и применять все это для нахождения оптимального решения.

И еще, желательно, чтобы в заданиях, хотя некоторых, просматривалось практическое использование в том или ином профессиональном направлении.

Олимпиада закончена, пора приступить к **оценке результатов**. Результаты выполнения заданий оцениваются в баллах. Максимальный балл указан в каждом задании, он устанавливается жюри при подготовке заданий. Если участник допустил ошибки или выбрал неоптимальное решение, задание будет оценено меньшим баллом. Кроме этого все задания разбиты на группы, по темам, и по каждой теме также подсчитываются сумма набранных баллов.

Победители олимпиады определяются по сумме набранных баллов в личном зачете, при этом нужно не только набрать максимальное количество баллов, но и выполнить задания из всех обязательных тем.

Призерами олимпиады могут стать первые три участника, набравшие максимальное число баллов.

В нашей олимпиаде имеется еще одна форма поощрения участников: студенты, успешно выполнившие задания по всем обязательным темам, и набравшие по ним определенное количество баллов, получают зачет или экзамен по курсу информатики. Именно эта форма поощрения позволяет поднять общий уровень подготовки по предмету, так как привлекает к участию в олимпиаде больше студентов и стимулирует самостоятельную подготовку. При подготовке к олимпиаде студенты могут получить в жюри перечень вопросов, которые будут рассмотрены в заданиях.

И, наконец, **подведение итогов**, момент немаловажный: результаты должны быть справедливы, своевременны и, отчасти, являются рекламой для проведения олимпиад в будущем.

Итоги олимпиады вывешиваются на кафедре и всех факультетах через неделю после тщательной проверки работ. Участники, занявшие первые три места, приглашаются на заседание кафедры информатики и информационных систем, где в торжественной обстановке их поздравляют и вручают подарки. Кроме этого они получают денежные премии из средств академии, что тоже неплохо для студенческого бюджета.

Результаты студентов, которым по итогам олимпиады удастся сдать экзамен по информатике, обсуждаются в группах во время занятий. Экзаменационная оценка выставляется во время сессии, но обязательным условием для ее получения является посещение лекций и выполнение всех практических работ.

М. О. Лебедев, кандидат технических наук, доцент

ПРИНЦИПЫ СОЗДАНИЯ ДИНАМИЧЕСКИХ ЖУРНАЛОВ ДЛЯ ИНФОРМАЦИОННО-ПОИСКОВЫХ И РЕГИСТРИРУЮЩИХ СИСТЕМ

В большинстве случаев информационно-поисковые и регистрирующие системы (ИПРС) могут быть представлены в виде совокупности трех составляющих:

1. основной список;
2. информационный блок;
3. журналы.

Под основным списком будем понимать информацию, параметры которой (состав полей) не меняется (или меняется крайне редко). Каждая запись основного списка определяет основную единицу хранения в информационной системе (ИС). Например, список студентов вуза, список учащихся школ, список пациентов больницы, список производимой продукции, список товаров и т.д. Основной список представляет собой высоко нормализованную (не ниже третьей нормальной формы) таблицу базы данных. Состав полей этой таблицы практически неизменяемый.

Под информационным блоком будем понимать некую совокупность параметров, описывающих каждый элемент основного списка. Состав параметров может изменяться при переходе от одного элемента основного списка к другому. Этот блок содержит актуальную информацию и не содержит историю изменения этой информации.

Под журналом будем понимать некую совокупность параметров, каждое изменение которой регистрируется как отдельная запись. Совокупность параметров связана с элементами основного списка. Состав полей каждого журнала определен, но может меняться в ходе развития информационной системы. Этот блок содержит историю (движение) параметров, описывающих каждый элемент основного списка.

Рассмотрим принципы создания клиентских приложений ИПРС, когда в их состав могут входить различные журналы, состав и структура данных которых не определен полностью на момент проектирования системы, или могут меняться в ходе эксплуатации системы.

Для определения состава журналов для данной ИПРС будем использовать таблицы, в которых будем регистрировать каждый новый журнал, указывать какие поля (их типы и способ ввода данных) будут входить в каждый журнал, а также таблицу для выбора значений для полей, способ ввода которых предполагает выбор значения из списка (как и прежде, полагаем, что выбор значения из списка может использоваться для типа дан-

ных "текстовый"). Структура и отношения этих таблиц показаны на рисунке ниже.

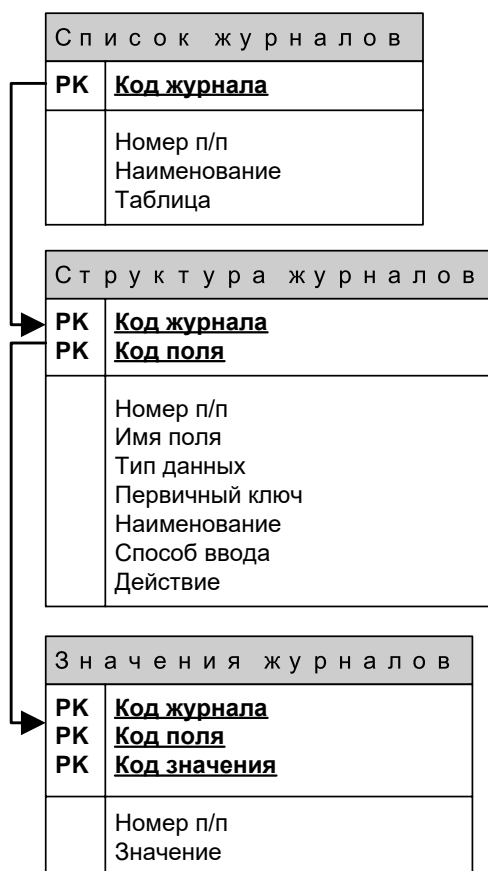


Рис. 1. Таблицы регистрации создания журналов

Поле "Наименование" (таблица "Список журналов") используется для отображения названия журнала на закладке страниц (если в ИС журналов несколько, то логично каждый журнал располагать на отдельной странице). Поле "Таблица" (таблица "Список журналов") содержит собственно имя таблицы, в которой будет храниться информация данного журнала.

Поле "Имя поля" (таблица "Структура журналов") определяет имя поля, как оно будет указано в таблице данного журнала. Поле "Тип данных" определяет тип данных создаваемого поля. Поле "Первичный ключ" определяет, входит ли данное поле в состав первичного ключа создаваемого журнала (в состав первичного ключа журналов в обязательном порядке будет входить как внешний ключ первичный ключ основного списка; полностью первичный ключ журналов образуется совокупностью первичного ключа основного списка и полей журнала, отмеченных, как входящих в первичный ключ). Поле "Наименование" используется для отображения соответствующей колонки в программе, работающей с данной базой дан-

ных. Поле "Способ ввода" аналогично такому же полю таблицы "Параметры". Поле "Действие" определяет какие действия необходимо выполнить при изменении (изменении значения, удалении значения) данного поля. В этом поле указывается имя метода (и при необходимости, имя программного компонента – dll-модуля, СОМ-сервера), который необходимо выполнить при изменении значения этого поля. Этим самым обеспечивается обработка событий при работе с компонентами ИС (третье требование динамически настраиваемых ИС). Поле "Номер п/п" во всех таблицах используется для сортировки записей при отображении информации. Таблица "Значения журналов" по своему назначению полностью идентична таблице "Значения параметров". На базе информации, хранящейся в этих системных таблицах, создаются таблицы-журналы, которые собственно и используются для хранения информации. Структура этих таблиц в обобщенном виде может быть представлена следующим образом

Журнал (таблица)	
<u><Поля для связи с основным списком></u>	РК
<u><Прочие поля первичного ключа></u>	РК
<Прочие поля>	

Рис. 2. Таблица-журнал – обобщенная структура

Эта таблица связана с основным списком (для этого используются "Поля для связи с основным списком", которые являются элементами первичного ключа журнала) и может быть связана с таблицей "Значения журналов", если какие-либо поля предполагают выбор значений из списка.

Каждый журнал представляет собой отдельную таблицу базы данных, состав которой определяется в специальной таблице "Структура журналов". Создание новой или модификация существующей таблицы (например, при изменении состава полей журнала) не является сложной задачей. Но изменение структуры существующей таблицы (ранее созданного журнала) или создание новой таблицы (нового журнала) влечет изменение интерфейса клиентских приложений, работающих с данной ИПРС. Динамически настраиваемые информационные системы должны настраивать пользовательский интерфейс при изменении структуры базы данных. Одним из вариантов решения динамически настраиваемого интерфейса журналов является создание, так называемого, "универсального интерфейса", когда для работы с каждым журналом (в клиентском приложении) используются одни и те же визуальные компоненты (таблицы, списки, деревья и т.д.). А доступ к каждому конкретному журналу осуществляется через визуальные компоненты управления (например, кнопки или закладки). В

этом случае при добавлении нового журнала элементы управления, осуществляющие доступ к конкретным журналам, создаются динамически при загрузке клиентского приложения, используя таблицу "Список журналов". Визуальные элементы для корректировки данных в журналах также создаются при загрузке приложения (при этом используется таблица "Структура журналов"). По характеру регистрируемой в журналах информации все журналы могут быть разделены на 2 группы: первичные и вторичные.

К первичным журналам относятся те, добавление и корректировка (но не удаление) информации в которые не зависит от состояния других журналов.

К вторичным относятся журналы, имеющие поля-ссылки на поля в первичных журналах.

Например, пусть один из журналов "Регистрация заявок клиентов", а другой журнал – "Меры по заявкам". Очевидно, что второй журнал должен иметь поле (или поля), идентифицирующую заявку, на которую были приняты меры. В журнал "Регистрация заявок клиентов" информация вносится по мере поступления заявок от клиентов. Новая заявка никак не зависит от мер, принятых по ранее сделанным заявкам. В свою очередь, при регистрации принятых мер необходимо указать, на какую именно заявку были приняты те или иные меры. Для регистрации принятых мер оператор должен иметь возможность выбора конкретной заявки из списка (причем, на этот список могут накладываться дополнительные условия, например, фильтрация по определенным параметрам и т.п.). Очевидно, что установка связи типа "Главная-подчиненный" между этими двумя журналами является недопустимой (необходимо видеть как все поданные заявки, так и все принятые меры по всем заявкам). Здесь решением может быть опять же "универсальный интерфейс" – создание единого визуального средства выбора значения из списка для любых подобных журналов. Этот универсальный интерфейс должен настраиваться под состав отображаемых в списке полей. Командные элементы интерфейса (кнопки) должны иметь универсальный алгоритм обработки щелчка (выбора элемента списка), параметрами которого являются наборы данных источника (откуда идет выбор) и приемника (куда помещается выбранное значение) и их поля.

В свою очередь, при удалении какой-либо записи из первичного журнала необходимо просматривать вторичные журналы и удалять из них записи, связанные с удаляемой записью из первичного журнала. Удаление же записей из вторичных журналов не затрагивает содержимого других журналов.

Таким образом, используя средства динамического создания таблиц баз данных и универсальные интерфейсы для отображения информации из

журналов и ввода ссылок на записи в первичных журналах, решаем задачу создания динамических информационных систем.

Н.В. Лушкин, кандидат технических наук, доцент

МОДЕЛЬ ЛЕСОХОЗЯЙСТВЕННОГО ПРОИЗВОДСТВА

Рассмотрим производство лесоматериалов, состоящее из k процессов работающих последовательно (заготовка древесины, вывоз древесины, распиловка, сушка, хранение).

Перечислим предположения, на которых основана модель [1].

1. Лесхоз осуществляет k -этапный производственный процесс, в котором производится различные пиломатериалы. Все без исключения единицы конечного продукта должны пройти все k стадии в заданном порядке и заказ может быть выполнен на стадии $j \leq k$.

2. Каждый процесс описывается своей собственной производственной функцией, не зависящей от производственных функций остальных $k-1$ процессов.

3. Выпуск продукции Q_j ($j=1, 2, \dots, k$) в единицу времени (темп производства) j -м процессом является случайной величиной. Функция плотности вероятностей темпа $f_j(q)$ полностью определяется уровнем использования производственных факторов в этом процессе в течение планового периода (ПП). Таким образом, изменяя распределение фактора производства, предприятие может менять вероятностное распределение величины Q_j . Если функция $f_j(q)$ определена, то также определены математическое ожидание $E(Q_j)$ и дисперсия $Var(Q_j)$ темпа производства j -го процесса.

4. Для описания выпуска в единицу времени Q_j удобно взять величину $ST_j=1/Q_j$, которая определяет время, необходимое для выпуска единицы продукта, или одного производственного заказа. Функция плотности вероятностей величины ST_j и ее параметры полностью определяются уровнем использования производственных факторов. Таким образом, ST_j – случайная величина с известной функцией плотности вероятностей $f_j(ST)$, математическим ожиданием ET_j и дисперсией VT_j . Другими словами, предприятие не может полностью управлять величиной ST_j , но может воздействовать на ET_j и VT_j или даже на вид функции $f_j(ST)$ изменением количества фактора, в j -м процессе. После распределения фактора величина ST_j становится случайной и больше не контролируется предприятием.

5. Число заказов, получаемых предприятием в единицу времени (или количество продукции, которую можно продать в единицу времени

по определенной цене после j -го процесса, $j \leq k$), является случайной величиной.

6. Считается, что, выбрав распределение производственных факторов, предприятие в течение периода планирования ПП принимает все заказы, даже если она окажется не в состоянии завершить (или даже начать) выпуск продукции по всем заказам в этом периоде.

7. В начале планового периода ПП руководство предприятия должно принять два различных решения: относительно уровней затрат на сбыт и относительно распределения факторов по k производственным процессам. Первое решение определяет $f(D)$, $E(D)$, $Var(D)$ на период ПП, а второе определяет $f_j(Q)$, $E(Q_j)$, $Var(Q_j)$, где $j=1,2,\dots,k$.

Перечислим переменные и уравнения модели, описывающую рассматриваемое предприятием процесс производство:

WT - среднее время ожидания заказа в очереди.

IDT - среднее время простоя системы в ожидании очередного заказа.

WT_{ij} - время обработки i -го заказа в j -м процессе, $i=1, 2, \dots, m$; $j=1,2,\dots,k$.

DT_{ij} – время простоя процесса j в ожидании i -го заказа.

AT_i – интервал времени между появлениями i -го и $(i+1)$ -го заказами.

ST_{ij} - время обработки заказа.

T_{ij}=WT_{ij}+DT_{ij} – полное время пребывания i -го заказа в j -м процессе.

f(AT) – функция распределения плотности вероятностей интервала времени между двумя последовательными заказами.

f(ST) - функция распределения плотности вероятностей времени обслуживания.

Считается, что в момент получения первого заказа, т.е. при $i=1$, производственный процесс описывается уравнениями $AT_1=0$, $TD_{11}=0$, $DT_{12}=ST_{11}$, ..., $DT_{1k}=\sum ST_{1j}$, $WT_{11}=0$, $WT_{12}=0$, ..., $WT_{1k}=0$, $T_{11}=ST_{11}$, $T_{12}=ST_{12}$, ..., $T_{1k}=ST_{1k}$.

При поступлении дальнейших заказов, т.е. при $i=2,3,\dots,m$, эти уравнения надо соответствующим образом изменить. Уравнения примут вид $T_{i1}=WT_{i1}+ST_{i1}$, $i=2, 3, \dots, m$, $T_{i2}=WT_{i2}+ST_{i2}$, $i=2, 3, \dots, m$, ... $T_{ik}=WT_{ik}+ST_{ik}$, $i=2, 3, \dots, m$,

Ожидает ли заказ очереди в данном процессе или, напротив, процесс простаивает, зависит от знака разностей $DIF_1=T_{i-11}-AT_i$

$DIF_2=(T_{i-11}+T_{i-12})-(AT_i+WT_{i1}+ST_{i1})$.

Если для j -го процесса $DIF_j > 0$, то время простоя процесса равно нулю, а время ожидания равно $WT_{ij} = DIF_j$. Если $DIF_j < 0$, то время ожидания процесса равно нулю, а время простоя равно $DT_{ij} = -DIF_j$. Если $DIF_j = 0$, то и время ожидания, и время простоя процесса равны нулю. Величина AT_i

считается случайной величиной с функцией плотности вероятностей $f(AT)$, математическим ожиданием ET и дисперсией VT . Для каждого процесса ST_{ij} считается случайной величиной с функцией плотности вероятностей $f_j(ST_j)$, математическим ожиданием ET_j и дисперсией VT_j .

Продолжительность планового периода равна шести месяцам ($ПП=180$);

Переменной реакции, или зависимой переменной, в рассматриваемом эксперименте считается прибыль. *Факторами* эксперимента являются затраты на производство и затраты на сбыт. Оба эти фактора – *количественные*. Таким образом, существует функциональная связь между затратами на производственные нужды, на сбыт и прибыльностью предприятия. Руководство предприятия теоретически может выбрать из бесконечного числа уровней каждого фактора, на практике приходится ограничиваться конечным числом уровней. Предположим, что предприятие рассматривает только 5 различных вариантов оперативного плана, каждый из которых содержит следующие показатели: 1) стратегию сбыта, 2) распределение ресурса по различным этапам производства (5 этапов), 3) полные затраты C .

Значения математического ожидания спроса $E(D)$ и темпов производства $E(Q_j)$ (в единицах в день), а также полных затрат C (в рублях) для модели предприятия представлены в таблице 1.

Т а б л и ц а 1

План	E(D)	E(Q _j)					C
		Заготовка	Вывоз	Распиловка	Сушка	Хранение	
1	4.00	4.25	-	-	-	-	1800
2	4.00	4.50	4.35	-	-	-	2000
3	4.50	5.00	4.75	4.65	-	-	3250
4	5.75	7.00	7.25	7.00	6.50	-	4550
5	7.75	9.00	9.50	9.25	8.50	8.50	5720

Используя систему Mathcad, были проведены эксперименты для n повторений (реплик) для каждого плана. Каждая реплика – это просчет модели в течении $ПП$ единиц времени. В результате каждого просчета вычисляется полная прибыль $ПРИБЫЛЬ = P_i * Q - C$, где P_i – цена, Q – число единиц продукции, произведенной в течение периода $ПП$, C – полные затраты. В течение всего эксперимента цена единицы конечной продукции постоянна. Имитировалась функционирование предприятия в течение шести месяцев и подсчитывалась полная прибыль за этот период. Полученная модель позволяет определить объемы заготовок леса и степень переработки пиломатериалов.

Библиографический список

1. *Нейлор Т.* Машинные имитационные эксперименты с моделями экономических систем. – М.: Мир, 1975; 500 с.

Н.В. Лушкин, кандидат технических наук, доцент

ИДЕНТИФИКАЦИЯ МАТЕМАТИЧЕСКОЙ МОДЕЛИ ПРОЦЕССА РАСТВОРЕНИЯ В КОЛОННОМ АППАРАТЕ

Одним из наиболее перспективных направлений в технологии выщелачивания как мелкодроблёных, так и крупнодроблёных бокситовых спёков является использование колонных аппаратов с пульсациями в подаче выщелачивающей жидкости. В условиях случайных возмущений концентрация алюминатного раствора и оптимальный температурный режим выщелачивания меняется случайным образом. Управление таким процессом потребовало разработки и идентификации математической модели. В [1] разработана математическая модель с распределёнными параметрами. Для идентификации математической модели и определения эффективности массообмена процесса выщелачивания, в противотоке, по высоте аппарата, были отобраны пробы раствора на отметках по высоте: 0; 1.2; 3.5; 6.0; 7.8; 10.6 м через установленные на корпусе аппарата пробковые краны. Пробы отбирались последовательно, начиная с верхней точки аппарата. Время отбора одной серии опытов 20- 25 мин. Расход жидкости на выщелачивание и промывку менялся в зависимости от заданного режима от 20 до 50 м³/час, количество перерабатываемого спёка от 5 до 34 т/час, уровень шлама также менялся в широких пределах. Результаты экспериментов обрабатывались по методу наименьших квадратов. При этом за основу структуры математической модели использовались уравнения [1] и по методу наименьших квадратов определялись параметры связей. Результаты расчётов приведены в таблице 1. Из таблицы видно, что при широком диапазоне изменений производительности колонного аппарата от 14 т/час до 33.8 т/час, концентрации полезных компонентов в растворе удовлетворительно совпадают расчётные и экспериментальные значения концентраций. Совпадение расчётных и экспериментальных концентраций в пробных точках свидетельствует о достоверности математической модели реальному процессу.

Модель [1]

$$\frac{\partial C_1}{\partial t} + V \frac{\partial C_1}{\partial Z} = D_p \frac{\partial^2 C_1}{\partial Z^2} + \frac{B_M}{\varepsilon} (C_s - C_1),$$

$$\frac{\partial C_1}{\partial t} - W \frac{\partial C_s}{\partial Z} = D_{ps} \frac{\partial^2 C_s}{\partial Z^2} + \frac{A_M (C_H - C_s) - B_M (C_s - C_1)}{(1 + \varphi_1 \eta / 2 - (1 - \eta / 2)(1 - \varepsilon)}$$

в стационарном случае была использована для определения концентрации продукционного алюминатного раствора на выходе из колонны. Преимущество полученной модели состоит в том, что, задавая среднюю концентрацию $C_s(0)$ на поверхности частиц на выходе их аппарата, т.е. задавая потери целевого компонента со шламом, можно определить концентрацию целевого компонента на выходе из колонны.

Или задавая допустимую концентрацию потерь целевого компонента со шламом $C_s(0)$, концентрацию продукционного раствора, состав спёка, можно определить проектные размеры аппарата и его производительность. Модель [1] была использована для аппроксимации распределения концентраций полезных компонентов в растворе по высоте аппарата. При этом средний радиус частиц 0.5 мм, скорость спёка вниз в аппарате $W = 0.0003$ м/с, скорость растворителя $V=0.01$ м/с, средняя порозность $\varepsilon= 0.5$. В итоге были вычислены значения коэффициентов $A(5)=-0.000022$, $D_p=0.015$, $D_{ps}=0.13$.

Результаты сопоставления физических и расчётных данных при найденных коэффициентах приведены в таблице 1. Из данных таблицы следует, что модель хорошо "откликается" на все основные параметры процесса, что и послужило основанием использовать её, с одной стороны для расчёта конструктивных параметров аппаратов, с другой - для управления процессом выщелачивания и промывки материалов в колонных аппаратах.

Т а б л и ц а 1

Концентрация полезных компонентов в растворе						
Раствор с отметки, м	Промышленный аппарат			Расчёт		
	Al_2O_3	Na_2O_T	Na_2O_K	Al_2O_3	Na_2O_T	Na_2O_K
7.8	57.1	54.5	51.1	57.7	60.3	55.7
6.0	27.5	32.2	29.1	20.0	35.1	23.2
3.5	8.7	16.0	11.7	6.3	15.3	9.1
1.2	2.9	11.9	9.2	2.4	10.1	6.5
0.0	1.6	9.4	5.9	1.8	9.4	5.9

Модель отмывки шлама

Разработка процесса выщелачивания для бокситового спёка в колонных аппаратах потребовала разработки модели процесса отмывки выщелоченного шлама.

Уравнения промывки выщелоченной частицы шлама для противоточного процесса имеют вид, по аналогии с процессом выщелачивания для

случая, когда целевой компонент отсутствует в твёрдом состоянии в частице, т.е. для процесса промывки коэффициент $A_M=0$. В этом случае уравнения [1] принимают вид:

$$(1) \quad \begin{aligned} \frac{\partial C_1}{\partial t} + V \frac{\partial C_1}{\partial Z} &= D_p \frac{\partial^2 C_1}{\partial Z^2} + \frac{B_M}{\varepsilon} (C_s - C_1), \\ \frac{\partial C_s}{\partial t} - W \frac{\partial C_s}{\partial Z} &= D_{ps} \frac{\partial^2 C_s}{\partial Z^2} - \frac{B_M (C_s - C_1)}{(1 - \varepsilon)} \end{aligned}$$

Для определения параметра B_M и скорости промывки бокситового шлама, были проведены исследования в лабораторных условиях. Исходя из условий проведения экспериментов можно считать, что $C_1=0$ так как шлам промывали водой, $W=0$, $D_{ps}=0$.

В этом случае уравнения будут иметь вид:

$$(2) \quad \frac{\partial C_s}{\partial t} = - \frac{B_M (C_s - C_1)}{(1 - \varepsilon)}.$$

Изменение массосодержания частиц во времени равно потоку вещества с поверхности частиц

$$(3) \quad \frac{\partial M}{\partial t} = 4\pi R^2 (m_t + m) D \left(\frac{\partial C}{\partial r} \right)_{r=R},$$

где R - радиус частиц шлама, m_t+m - пористость частиц шлама, D - коэффициент диффузии.

Будем считать, что концентрация раствора внутри частицы распределена равномерно и равна C_s . Граничные условия к уравнениям (2) и (3) служат условия на поверхности частицы

$$(4) \quad (m_t + m) D \left(\frac{\partial C}{\partial r} \right)_{r=R} = k (C_s - C_{10})$$

где k - параметр, необходимый определению, $C_{10}=0$.

Концентрацию полезного компонента внутри частицы можно вычислить по формуле

$$C_s = \frac{M}{(m_t - m)V},$$

где V - объём частицы, M - масса полезного компонента в частице.

Тогда уравнение (2) с учётом (4) запишем в виде:

$$(m_t - m)V \frac{\partial C_s}{\partial t} = -4\pi R^2 k C_s. \quad (5)$$

Полученное уравнение легко интегрируется, получаем

$$C_s = k_2 e^{-4\pi R^2 kt / V(m_t - m)}. \quad (6)$$

Очевидно, что в начале процесса $C_s = C_{H0}$, где C_{H0} - средняя концентрация раствора в начальный момент внутри частицы.

Для случая наших экспериментов $k_2 = C_{H0}$. Тогда изменение концентрации полезных компонентов в шламе для нашего случая будет

$$C_s = C_{H0} e^{-4\pi R^2 kt / V(m_t - m)}. \quad (7)$$

В уравнении (7) неизвестным является параметр k . Обработав экспериментальные данные с использованием критерия наименьших квадратов, найдём постоянный коэффициент промывки бокситовых шламов

$$\bar{k} = k / (m_t - m) = 0.25 \cdot 10^{-5}.$$

Объём частицы радиуса R равен

$$V = 4/3\pi R^3.$$

Тогда соотношение (7) запишется в виде

$$C_s = C_{H0} e^{-7.5 \cdot 10^{-6} t / R}.$$

Используя соотношения (2) и (5) определим параметр B_M

$$B_M = 7.5 \cdot 10^{-6} t(1 - \varepsilon) / R. \quad (8)$$

Полученная модель процесса промывки шламов с коэффициентом B_M , была использована в стационарном случае для определения значений концентраций полезных компонентов в растворе и в шламе по высоте колонного аппарата. Скорость движения шлама $W = 0.003$ м/с, скорость движения растворителя $V = 0.01$ м/с, порозность слоя $\varepsilon = 0.5$, при этом использовался шлам крупностью $R_1 = 2$ мм, $R_2 = 2.7$ мм, $R_3 = 3$ мм, $R_4 = 3.3$ мм, $R_5 = 3.7$ мм, $R_6 = 5.9$ мм.

На основании этих данных была рассчитана величина недоизвлечения за счёт недоотмывки из шлама по высоте аппарата по формуле

$$y = C_s \frac{m_t + m}{m},$$

где γ – плотность шлама, n - содержание Al_2O_3 в спёке, доли единиц.

Полученная модель позволяет определить высоту аппарата с учётом требуемой степени отмывки для материала заданной крупностью. При известной высоте аппарата может быть решена задача определения предельной крупности перерабатываемого материала. В тоже время полученная модель может быть использована для создания системы управления процессом выщелачивания в аппаратах колонного типа. В результате исследований доказано, что для обеспечения высокого извлечения полезных компонентов из бокситовых спеков необходимо учитывать кинетику отмывки раствора из спор спека и при расчете размеров аппарата для выщелачивания необходимо дополнительное время для промывки. При выборе и расчете конструкций колонных аппаратов, вместе с заданной производительностью, необходимо учитывать разнообразные технологические ограничения, диктуемые особенностями обрабатываемого материала, требованиями к параметрам продукционного раствора и ограничениями на потери полезных компонентов.

Библиографический список

1. Лушкин Н.В. Актуальные проблемы современной математики. Т. 2. 2004, сс.100-108.
2. Протодьяконов И.О., Люблинская И.Е., Рыжков А.С. Гидродинамика и массообмен в дисперсных системах жидкость - твердое тело. - Л.: Химия, 1987; 336 с.
3. Аксельруд Г.А. Теоретические основы химической технологии, 1953, № 5, сс.674-677.
4. Каузов П.А. Основы анализа дисперсного состава промышленных пылей и измельченных материалов. - М.: Химия, 1971; 283 с.
5. Лушкин Н.В., Абрамов В.Я., Иванов В.А. Управление распределёнными процессами в колонных аппаратах с плотно движущимся слоем. //Изв.вуз.СССР, разд."Цветная металлургия". 1990.№ 5 с.102-107.
6. Абрамов В.Я., Лушкин Н.В. Модель процесса выщелачивания полидисперсных бокситовых спеков.//Цветная металлургия. 1990.№ 1 с.8-10.

АНАЛИЗ ПОСЕЩАЕМОСТИ СТУДЕНТАМИ САЙТА ПРЕПОДАВАТЕЛЯ В ПРОЦЕССЕ ОБУЧЕНИЯ

Уже в течение почти десяти лет я использую Интернет при обучении студентов Санкт-Петербургской лесотехнической академии. С этой целью был разработан и размещен в Интернете личный сайт, который может оказать студентам методическую поддержку по читаемым мною курсам. На этом сайте (www.habarov.spb.ru) размещены конспекты лекций в виде текстовых архивов, презентаций или html-документов, описания лабораторных работ и индивидуальных заданий, а также в период сессии там размещаются вопросы экзаменационных тестов.

Однако в этой статье я не буду останавливаться на методических аспектах представленной на сайте информации и эффективности ее использования, а хотел бы поделиться теми результатами, которые были получены из анализа статистики посещений моего сайта в Интернет. При этом анализ проводился как с технической, так и с временной точек зрения.

Технический аспект позволяет определить требования к сайту, как Интернет-ресурсу с учетом его группы пользователей. Временной аспект, а именно семестр, сессия и каникулы - позволяет определить загрузку сайта, регулярность и интенсивность работы студентов по тем или иным дисциплинам. Анализ выполнен на базе осеннего семестра 2008 года и сессии января 2009 года. В этом семестре я вел три потока студентов совершенно различной направленности:

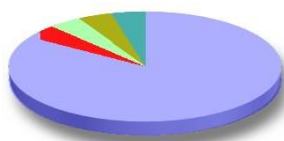
- поток студентов-экономистов 2 курса - предмет "Информатика" (~100 человек),
- поток информационщиков 3 курса - "Представление знаний в ИС" (~25 человек),
- поток информационщиков 4 курса - "Информационные сети" (~25 человек).

Таким образом, общее количество обучаемых студентов было около 150 человек. Все они были ориентированы на работу с сайтом. Это было связано с принятой мною в этом семестре методикой, при которой все лекции читались на базе презентаций, что позволяло не только наглядно представить информацию, но и увеличить ее объем в каждой из лекций. Данный подход не позволял студентам вести достаточно полные тетрадные конспекты, но при этом полные конспекты и презентации по каждой лекции им были доступны на сайте лектора. Они могли их просматривать че-

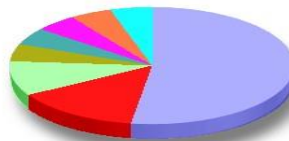
рез Интернет или копировать на свой компьютер и использовать при подготовке к экзаменам по вышеназванным предметам.

Анализ статистики посещений сайта начнем с момента окончания сессии и всех дополнительных пересдач, то есть с недели каникул. Так как сайт является общедоступным в Интернете, то это означает, что его пользователями могут являться не только обучающиеся студенты ЛТА, но и любые пользователи Интернет. Эта статистика нам необходима, чтобы определить фоновую группу пользователей сайта.

География по странам и городам (с 01.02.2009 по 07.02.2009)



Россия – 21,
Казахстан – 1,
Узбекистан – 1,
Германия – 1,
США – 1



Санкт-Петербург – 11,
Москва – 3,
Владивосток – 2,
Мурманск – 1,
Новосибирск – 1,
Смоленск – 1,
Краснодар – 1,
Другие – 3

Из приведенных графиков видно, что за первую неделю февраля суммарная группа пользователей из Санкт-Петербурга составила всего 11 человек, то есть менее, чем по 2 человека в день. Но если проанализировать данные за предыдущий месяц, а это был январь 2009 года, то есть сессия, то мы получим совершенно другие данные.

География по странам (с 05.01.2009 по 30.01.2009)

Страна	Посетители		Хиты		Загрузки		Хиты/посет.	Загрузки/посет.
Россия	251	95,1%	479	95,6%	568	96,1%	1,91	2,26
Германия	3	1,1%	3	0,6%	3	0,5%	1,00	1,00
Украина	2	0,8%	8	1,6%	9	1,5%	4,00	4,50
Европа (не опред.)	2	0,8%	4	0,8%	4	0,7%	2,00	2,00
Другие	6	2,2%	7	1,4%	7	1,2%	1,20	1,20
ВСЕГО В ОТЧЕТЕ:	264	100,0%	501	100,0%	591	100,0%		

В группу другие входят: Беларусь, Франция, Киргизия, Латвия, Норвегия, Швеция.

В группу другие входят: Новочеркасск, Екатеринбург, Шелково, Хабаровск, Ставрополь, Белгород, Вологда, Томск, Киров, Пенза.

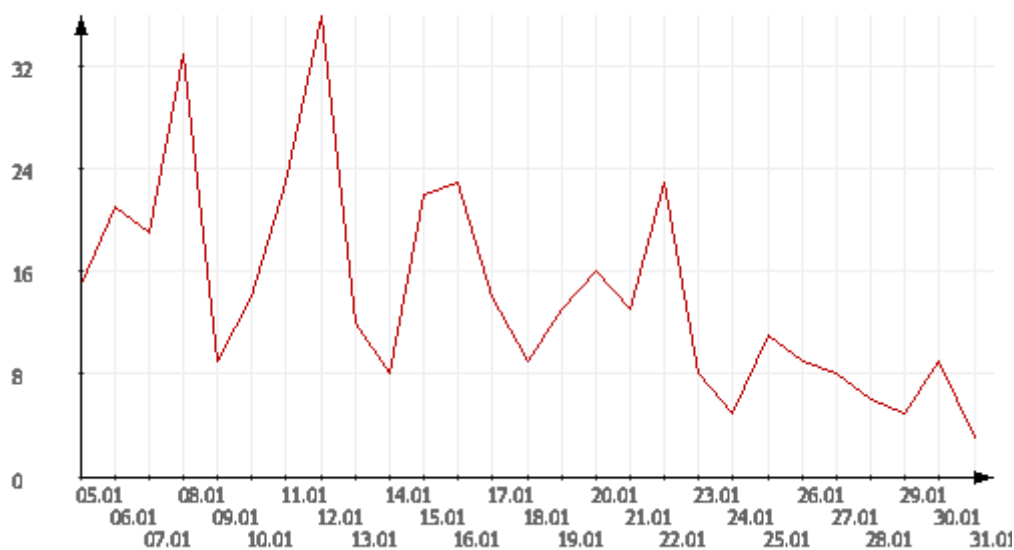
Из этих таблиц уже видно, что количество пользователей из Санкт-Петербурга существенно возросло и за 25 дней января составило 201 человек. Появляется надежда, что среди этой группы есть кто-то из тех 150 человек, которые пришли на сайт, чтобы почитать лекции или хотя бы узнать, когда и где они сдают экзамен и скачать вопросы к экзамену.

География по городам (с 05.01.2009 по 30.01.2009)

Город	Посетители		Хиты		Загрузки		Хиты/посет.	Загрузки/посет.
Санкт-Петербург	201	80,1%	399	83,3%	473	83,3%	1,99	2,35
Москва	13	5,2%	16	3,3%	18	3,2%	1,23	1,38
Иркутск	4	1,6%	4	0,8%	4	0,7%	1,00	1,00
Воронеж	3	1,2%	3	0,6%	3	0,5%	1,00	1,00
Ростов-на-Дону	2	0,8%	6	1,3%	13	2,3%	3,00	6,50
Красноярск	2	0,8%	3	0,6%	3	0,5%	1,50	1,50
Новосибирск	2	0,8%	2	0,4%	2	0,4%	1,00	1,00
Волгоград	2	0,8%	2	0,4%	2	0,4%	1,00	1,00
Краснодар	2	0,8%	20	4,2%	26	4,6%	10,00	13,00
Гатчина	2	0,8%	2	0,4%	2	0,4%	1,00	1,00
Саратов	2	0,8%	3	0,6%	3	0,5%	1,50	1,50
Казань	2	0,8%	3	0,6%	3	0,5%	1,50	1,50
Псков	2	0,8%	2	0,4%	2	0,4%	1,00	1,00
Уфа	2	0,8%	2	0,4%	2	0,4%	1,00	1,00
Другие	10	4%	12	2,5%	12	2,1%		
ВСЕГО В ОТЧЕТЕ:	251	100,0%	479	100,0%	568	100,0%		
из них на странице:	251	100,0%	479	100,0%	568	100,0%		

Преподавателю, потратившему столько сил и времени на разработку сайта и материалов для него было бы приятно узнать и осознать то, что он используется его студентами и помогает им при подготовке к экзаменам. Но чтобы можно было бы ответить на этот вопрос, давайте проанализируем распределение посетителей сайта по отдельным датам января 2009 года

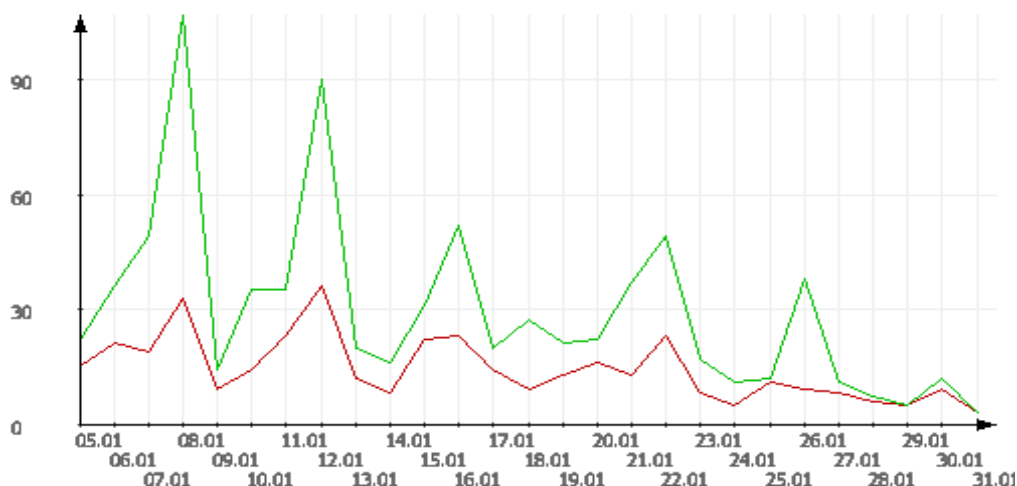
Посетители сайта (с 05.01.2009 по 30.01.2009)



На первый взгляд, трудно получить из этого графика какую-либо информацию. Но если принять во внимание тот факт, что экзамен по "Информатике" - 09.01.2009 (~100 человек), "Информационные сети" - 13.01.2009 (~25 человек) и "Представление знаний в ИС" - 17.01.09 (~25 человек), то график сразу оживает и может дать много полезной информации. Характерными являются также пики графика в районе 23.01.09 и 26.01.09 – дней первой и второй пересдач. Понятен и спад после 30.01.09, когда начались каникулы.

Таким образом, не вызывает сомнения, что январская аудитория посетителей сайта в основной своей массе представляет именно студентов ЛТА, обучающихся у автора сайта. А если это так, то попробуем проанализировать этот график. Настораживает тот факт, что 09.01.2009 сдают экзамен ~100 человек, а в дни до этой даты число посетителей много меньше. Радует то, что 13.01.2009 сдают экзамен ~25 человек, а в период с 9 до 13 января посетителей сайта не менее 12 человек. Если теперь совместить эти данные с оценками за экзамен, то корреляция явная. Второй поток сдал экзамен без двоек с преобладанием оценок 4 и 5, а первый имел 25% двоек, несколько 4 и 5, а в основном 3. Статистика сайта позволяет также отслеживать не только пользователей, но и интенсивность их работы.

Посетители сайта и хиты (с 05.01.2009 по 30.01.2009)

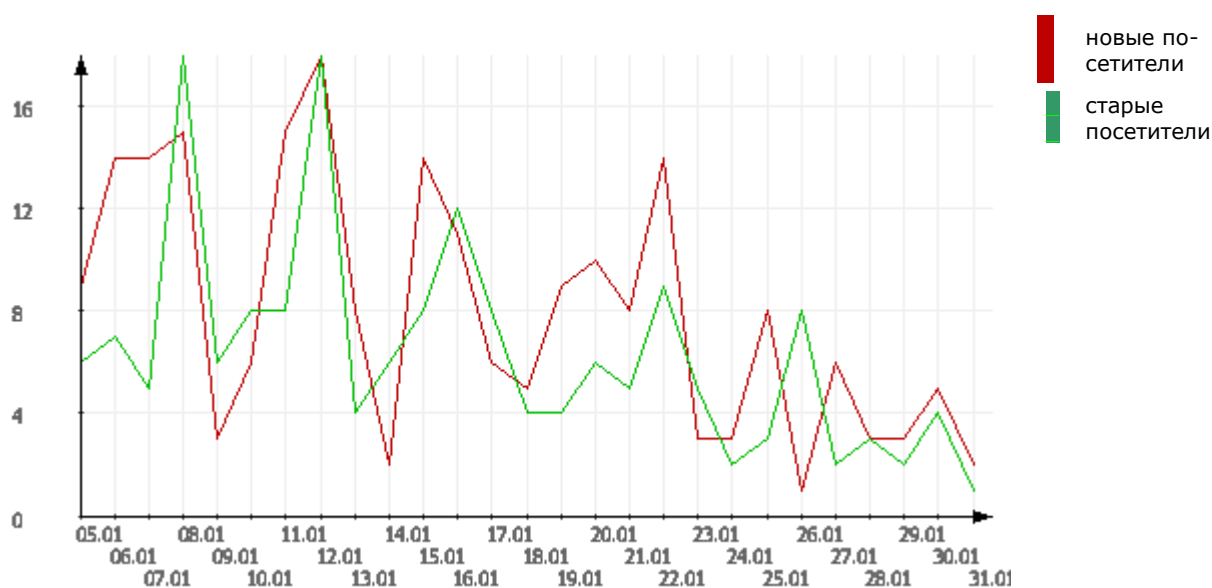


Любой преподаватель знает, что во многом результат обучения зависит от систематичности занятий. Приведенный ниже график иллюстрирует количество посетителей сайта в период сессии, которые до ее начала уже знакомы с материалами сайта (старые) или только первый раз зашли на этот сайт (новые).

Поток №1 показал на экзамене довольно средний уровень знаний. Но у этого же потока наблюдается и наиболее большая группа новых пользователей в период с 5 по 8 января. То есть большая часть студентов интенсивную подготовку к экзамену начали уже во время сессии.

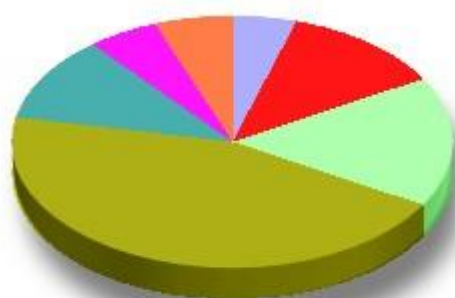
Показательным является и резкое увеличение новых пользователей после 17.01.09 – дня окончания сессии. Это те двоечники и неявившиеся, которым предстоят повторные пересдачи.

Старые и новые посетители сайта (с 05.01.2009 по 30.01.2009)



Важным для анализа посетителей сайта за заданный период является не только то – новый он посетитель или старый, а насколько интенсивно пользуется он методической информацией, представленной ему для подготовки к экзамену. При анализе статистики сайта эту информацию можно получить из отчета по частоте посещений.

Частота посещений сайта посетителями (с 05.01.2009 по 30.01.2009)



1-2 сессий в месяц	9	4,7%
2-4 сессий в месяц	23	11,9%
1-2 сессий в неделю	32	16,6%
2-4 сессий в неделю	86	44,6%
4-5 сессий в неделю	21	10,9%
5-7 сессий в неделю	10	5,2%
более 7 сессий в неделю	11	5,7%

Статистика сайта "Лояльность аудитории" позволяет получить распределение посетителей сайта по числу сессий за время его знакомства с сайтом. Из приведенной ниже таблицы видно, что почти половина студентов заходила на сайт, чтобы однократно скачать ту или иную лекцию или список вопросов к экзамену.

Вместе с тем существует и большая доля посетителей, которые имели более длительное знакомство с материалами сайта. И, как это ни странно, эти количества посетителей очень близко совпадают с распределением положительных оценок за экзамен во всех трёх потоках.

Лояльность аудитории (с 05.01.2009 по 30.01.2009)

Число сессий за время знакомства с сайтом	Посетители	
1 сессия	187	49,2%
2 сессии	46	12,1%
3-5 сессий	59	15,5%
6-10 сессий	55	14,5%
11-20 сессий	21	5,5%
21-50 сессий	12	3,2%
ВСЕГО В ОТЧЕТЕ:	380	100,0%

Отчет "Время знакомства с ресурсом" используется для определения среднего времени, в течение которого пользователи задерживаются на сайте. Эта информация показывает, сколько долго пользователь изучает тему, чтобы принять решение. В зависимости от этой информации, можно планировать эффект от работы сайта, как время задержки в принятии решения.

Время знакомства с ресурсом (с 05.01.2009 по 30.01.2009)

Время знакомства	Посетители	
менее суток	25	6,6%
1-2 дня	211	55,5%
2-4 дня	20	5,3%
4-7 дней	16	4,2%
1-2 недели	16	4,2%
2-3 недели	17	4,5%
3-4 недели	20	5,3%
4-8 недель	33	8,7%
8-14 недель	18	4,7%
более 14-и недель	4	1,1%
ВСЕГО В ОТЧЕТЕ:	380	100,0%

Если считать удовлетворительным, хотя бы месячное или более изучение курса, то видно, что порядка 50 посетителей составляли ядро сайта, а 18 посетителей, знакомых с ресурсом более месяца, очень сильно совпадает с количеством отличных оценок на экзаменах в период сессии.

Статистика "Возвраты на сайт" позволяет наряду с рядом полезной информации ответить и на такой вопрос, а зачем студенту в период сессии, когда у него на подготовку к экзамену по 3-4 дня, возвращаться на сайт через 7 и более дней. Ответ может быть только один – он либо не дошел до экзамена, либо получил два и его ждет пересдача. И, как показали результаты сессии, статистика сайта близка к числу пересдач в дополнительную сессию.

Возвраты на сайт (с 05.01.2009 по 30.01.2009)

Повторные посещения	Сессии	
возврат менее чем через 1 час	30	10,5%
возврат через 1-3 часа	53	18,5%
возврат через 3-8 часов	37	12,9%
возврат через 8-24 часа	55	19,2%
возврат через 1-3 дня	45	15,7%
возврат через 3-7 дней	23	8,0%
возврат через 7-30 дней	35	12,2%
возврат более чем через 30 дней	9	3,1%

Также на основании данной таблице можно выделить две группы студентов с различным подходом в подготовке к экзамену. Одна из этих групп очень интенсивно использует информационный ресурс в течение короткого интервала времени. Другая группа периодически и регулярно использует информационный ресурс в течение всех дней, выделенных на подготовку к экзамену.

Из этих данных можно предположить, что первая группа выкачивает данные к себе на локальный компьютер и оперирует с ними в автономном режиме. Другая группа предпочитает работать с методической информацией непосредственно через Интернет.

Подводя интегральный итог количественному и временному анализу по посещаемости сайта студентами во время экзаменационной сессии, можно отметить некоторые обобщенные параметры, которые обычно представляются в карточке посещаемости сайта.

Карточка посещаемости сайта (с 05.01.2009 по 30.01.2009)

Посещаемость		Активность посетителей	
Посетителей	280	Сессий на посетителя	1,96
Хостов	299	Хитов на посетителя	3,12
Сессий	549	Загрузок на посетителя	3,63
Хитов	873	Посетителей на хост	0,94
Загрузок	1017		

Объем аудитории		Характеристики сессии	
Ядро аудитории	55	Максимальное время сессии (мин)	557,23
Недельная активная аудитория	7	Среднее время сессии (мин)	10,33
		Максимальная длина сессии (хитов)	40
Посетителей на сайте одновременно		Средняя длина сессии (хитов)	1,59
Максимум	3		
В среднем	1		

Минимальные и максимальные показатели посещаемости сайта

	по часам		по дням		по неделям	
Посетители - MIN	30.01.2009 21:00 - 21:59	0	31.01.2009	3	26.01.2009 - 01.02.2009	37
Хиты - MIN	05.01.2009 0:00 - 0:59	1	31.01.2009	3	26.01.2009 - 01.02.2009	85
Посетители - MAX	12.01.2009 16:00 - 16:59	10	12.01.2009	36	05.01.2009 - 11.01.2009	103
Хиты - MAX	26.01.2009 22:00 - 22:59	24	08.01.2009	107	05.01.2009 - 11.01.2009	298

Здесь мы уже подходим к техническому аспекту анализа, который позволяет определить требования к сайту, как Интернет-ресурсу с учетом группы его пользователей. Как следует из приведенных выше таблиц, требование к серверу, на котором размещен сайт, минимальны, так как пиковая нагрузка на него не превосходит трех одновременных подключений.

Какие же требования должны быть к программному обеспечению, структуре и дизайну сайта? На этот вопрос также дает ответ анализ статистики сайта, позволяющий определять технические характеристики, подключаемых клиентов исследуемой группы посетителей.

Браузеры и операционные системы посетителей (с 5.01.2009 по 30.01.2009)

Операционная система	Посетители		Браузеры	Посетители	
Windows XP	298	77,0%	Microsoft Internet Explorer 7.0	102	26,4%
Windows Vista	74	19,1%	FireFox 3.x	92	23,8%
Windows NT	4	1,0%	Opera 9.5 - 9.63	89	23,1%
n/a	4	1,0%	Microsoft Internet Explorer 6.0	54	14,0%
Linux i686	2	0,5%	Opera 9.2x	24	6,2%
Windows CE	2	0,5%	FireFox 2.x	8	2,1%
			Safari 3.x	4	1,0%
ВСЕГО В ОТЧЕТЕ:	387	100,0%	ВСЕГО В ОТЧЕТЕ:	387	100,0%
из них на странице:	384	99,0%	из них на странице:	373	96,6%

Дисплеи посетителей (с 05.01.2009 по 30.01.2009)

Разрешение	Глубина цвета	Посетители	
1280x1024	32bpp	203	52,9%
1024x768	32bpp	96	25,0%
1024x768	16bpp	19	4,9%
1152x864	32bpp	19	4,9%
1440x900	32bpp	16	4,2%
1680x1050	32bpp	11	2,9%
800x600	32bpp	9	2,3%
176x220	16bpp	2	0,5%
234x282	32bpp	2	0,5%
800x600	24bpp	1	0,3%
800x600	16bpp	1	0,3%
1024x768	24bpp	1	0,3%
1280x1024	16bpp	1	0,3%
1440x900	24bpp	1	0,3%
1088x612	32bpp	1	0,3%
234x320	32bpp	1	0,3%
ВСЕГО В ОТЧЕТЕ:		384	100,0%

Java, JavaScript и Cookie посетителей (с 05.01.2009 по 30.01.2009)

JavaScript	Java	Cookie	Посетители	
1.3	вкл	вкл	280	72,4%
1.3	выкл	вкл	104	26,9%
- Не определено -	выкл	выкл	3	0,8%

Из анализа этих таблиц следует, что посетители сайта наряду с Windows XP широко используют Windows Vista. Однако начинают находить применение и Linux и Windows CE, что говорит о группе посетителей, которые используют для доступа к сайту КПК, коммуникаторы или телефоны. Это видно и из характеристик дисплеев, где наряду с форматом 800x600, появились форматы 234x320, 234x282 и 176x220. Это требует новых подходов к организации дизайна сайта. С точки зрения использования на сайте процедур и функций JavaScript никаких сложностей у посетителей возникать не должно, так как практически у всех пользователей имеется поддержка JavaScript версии 1.3.

Хосты наиболее активных посетителей (с 05.01.2009 по 30.01.2009)

IP-адрес (хост)	Хиты	
85.249.146.17 (17-146-249-85-dial-up.master.ru)	76	9,6%
92.100.80.229 (ppp92-100-80-229.pppoe.avangarddsl.ru)	26	3,3%
217.147.0.13	20	2,5%
84.52.73.191 (191.73.co-location.westcall.net)	18	2,3%
92.101.58.253 (ppp92-101-58-253.pppoe.avangarddsl.ru)	18	2,3%
87.249.59.162 (162.as-59.nienschanz.ru)	17	2,1%
92.255.81.74	16	2,0%
91.196.252.1 (c1n252.polustrovo.net)	15	1,9%
93.100.141.25	13	1,6%
195.189.80.36	12	1,5%
92.101.52.175 (ppp92-101-52-175.pppoe.avangarddsl.ru)	12	1,5%
92.100.73.129 (ppp92-100-73-129.pppoe.avangarddsl.ru)	12	1,5%
195.64.140.18 (ip195-64-140-18.users.dmnet.ru)	12	1,5%
85.249.146.129 (129-146-249-85-dial-up.master.ru)	11	1,4%
77.241.45.99	10	1,3%
ВСЕГО В ОТЧЕТЕ:	793	100,0%
из них на странице:	288	563,2%

Пример анализа подключения с ряда хостов (с 05.01.2009 по 30.01.2009)

Дата – хост 217.147.0.13	Хиты	
06.01.2009	2	0,3%
08.01.2009	18	2,3%
Дата – хост 84.52.73.191 (191.73.co-location.westcall.net)'	Хиты	
08.01.2009	16	2,0%
09.01.2009	2	0,3%
Дата – хост 92.101.58.253 (ppp92-101-58-253.pppoe.avangarddsl.ru)	Хиты	
18.01.2009	18	2,3%
Дата – хост 87.249.59.162 (162.as-59.nienschanz.ru)	Хиты	
20.01.2009	3	0,4%
21.01.2009	9	1,1%
22.01.2009	5	0,6%

Анализ хостов посетителей сайта показывает, что большинство из них подключаются через ADSL соединения или непосредственно к провайдерам домашних сетей, то есть имеют высокоскоростной доступ к Интернет. В этих условиях у посетителей не должно возникать никаких сложностей по доступу к сайту, даже при загрузке flash-роликов.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ ВЕДЕНИЯ ВЗАИМООТНОШЕНИЙ С КЛИЕНТАМИ В СИСТЕМЕ «1С:ПРЕДПРИЯТИЕ 8»

Статистические исследования, проведенные в экономически развитых странах, свидетельствуют о том, что [1]:

1. Затраты на привлечение нового клиента в пять раз больше, чем на удержание существующего клиента.
2. Заключить сделку с уже имеющимся клиентом существенно легче и в 5-10 раз дешевле, чем с новым покупателем.
3. Увеличение числа постоянных клиентов на 5% увеличивает объемы продаж более чем на 25%, а прибыль – на 50% - 100%.

Приведенные выше сведения, при всей их усредненности, красноречиво говорят о необходимости приложения максимума усилий на удержание уже существующих клиентов компании. Тем не менее, до сих пор многие компании основное внимание уделяют привлечению новых клиентов, реализуя высокочрезвычайно затратные маркетинговые программы.

В соответствии с теорией маркетинга [1], в процессе взаимоотношений фирмы с клиентами можно выделить три этапа: привлечение, удержание и лояльность. Под *лояльностью клиентов*, понимается комплекс параметров, характеризующих их поведение (объем и постоянство закупок) и восприятие поставщика товаров и услуг [1]. По крайней мере, два последних этапа сопряжены с формированием так называемой клиентской базы предприятия. Очевидно, что клиентская база – это важнейший актив компании, которым надо эффективно управлять. Под управлением, в частности, понимается:

- 1) Персонализация взаимоотношений с клиентом на основе изучения его предпочтений и поведения
- 2) Создание и поддержание у клиентов чувства приверженности выбранной торговой марки, а следовательно и приверженности поставщику этих товаров и услуг
- 3) Выявление наиболее перспективных клиентов с точки зрения долгосрочного сотрудничества

Очевидно, что успешное решение этих задач требует накопления и обработки больших объемов информации. Поэтому совершенно естественно встает вопрос о применении современных информационных технологий для решения этой задачи.

Одной из стратегий активизации долговременных отношений с существующими, а также потенциальными клиентами является концепция CRM (Customer Relationship Management – Управление Взаимоотношениями с Клиентами) [1,2]. Согласно этой концепции предполагается использование информационных технологий для сбора данных о клиентах на всех стадиях взаимоотношений с ними (привлечение, удержание, лояльность), последующего извлечения знаний из полученных данных, которые затем используются для выстраивания взаимовыгодных отношений с ними.

С точки зрения информационных технологий CRM – система представляет собой набор приложений, связанных единой бизнес – логикой и интегрированных в корпоративную информационную среду компании на основе единой базы данных. Используемое при этом программное обеспечение позволяет автоматизировать эти бизнес – процессы при осуществлении маркетинговых акций, продажах и обслуживании.

По данным из различных источников, использование этих систем позволяет, в частности [2]:

- Снизить административные издержки на 10-20%.
- Увеличить объемы продаж на 10- 30% в год на одного менеджера.
- Увеличить число заключаемых сделок на 5 – 15%.

При этом следует отметить, что срок окупаемости CRM – систем, в среднем, приблизительно равен одному году.

Основным преимуществом, получаемым в результате внедрения CRM – систем является возможность *персонализации* обслуживания клиентов, что приводит к общему увеличению объема продаж компании. Кроме того, CRM – система обеспечивает координацию действий различных подразделений фирмы, обеспечивая их единой программной платформой для взаимодействия с клиентами, уменьшая при этом типичные ошибки сотрудников типа «забыл позвонить» и «перепутал номер заказа».

Итак, к основным функциям, которые должны быть реализованы в CRM – системах относятся:

- сбор, накопление и хранение данных о клиентах компании, истории взаимоотношений с ними;
- анализ накопленных данных с целью выработки плана маркетинговых мероприятий и акций в интересующих компанию разрезах и формах представления;
- поддержка коллективной работы пользователей компании с данными о клиентах.

В соответствии с этим перечнем можно выделить следующие блоки, составляющие CRM – систему, а именно: *оперативный, аналитический и коллаборативный* [1,2,3].

Оперативный блок CRM – системы должен обеспечивать ввод и накопление данных о клиентах компании и истории взаимоотношений с ними. Эти данные могут вводиться как сотрудниками компании, так и самими пользователями, например, через Web – сайт компании. При этом должна быть предусмотрена возможность ведения единого массива данных, доступного для любого пользователя, находящегося в любом из офисов компании, вносящего изменения в этот массив в режиме транзакции. Иными словами в общем случае CRM – система имеет распределенную информационную структуру.

Другой важной функцией оперативного блока CRM – системы является обеспечение оперативного доступа к накапливаемым данным всех пользователей системы. При этом этот блок системы должен выступать в качестве органайзера, осуществляющего планирование работы менеджера, подсказывающего ему, когда и какие действия по работе с клиентом он должен предпринять.

Аналитический блок CRM – системы предназначен для проведения анализа данных, характеризующих взаимодействие компании с отдельными клиентами и их группами, выделенными по различным признакам. Одной из важнейших функций аналитического блока является дифференциация клиентов, построенная по статистическим данным прошлых периодов, содержащих сведения о частоте и виде взаимодействия с клиентами и их изменениях во времени. Это нужно для формирования, так называемого профиля клиента (или их групп), выделенных по различным признакам, а также для ранжирования клиентов с точки зрения их привлекательности для компании. Эта информация в дальнейшем используется для выработки целевых предложений, а также предоставления особых условий сотрудничества наиболее лояльным и выгодным клиентам. Еще одной функцией этого блока является формирование сведений о работе менеджеров компании, которые могут быть использованы для сравнения их работы и расчета поощрений.

Основной функцией коллаборативного блока CRM – системы является организация совместной работы нескольких подразделений компании при взаимодействии с клиентами. Эта работа базируется на использовании единой базы данных о взаимоотношениях с клиентами. В состав этого блока могут входить средства поддержки решения задач взаимодействия с клиентами, например, организация совместных маркетинговых мероприятий или средства доступа клиентов к информации о состоянии исполнения их заказов (например, через Интернет).

Из сказанного следует, что основная часть данных, необходимых для анализа взаимоотношений с клиентами, может быть получена из баз данных учетных модулей информационной системы предприятия. Поэтому

CRM – система должна рассматриваться как составная часть информационной системы управления предприятием, а используемое программное обеспечение иметь развитые средства интеграции в другие, используемые на предприятии программно – аппаратные комплексы.

В настоящее время на рынках экономически развитых стран и в России предлагается большое число программных продуктов, предназначенных для автоматизации решения задач управления взаимоотношениями с клиентами. Они существенно различаются по охвату реализованных функций, стоимости и другим параметрам. Стоимость лицензии на одно рабочее место в зависимости от комплектации, без учета затрат на компьютеры и на линии связи, может достигать у некоторых западных «брендов» до 7000\$ [1]. При выборе программного продукта необходимо учитывать множество факторов, к числу которых относятся:

- соответствие функциональной полноты программного продукта потребностям предприятия;
- возможность интеграции с существующими программными комплексами;
- возможность настройки на существующие на предприятии бизнес – процессы без их радикальной реорганизации;
- стоимость приобретения, внедрения и сопровождения программного продукта;
- надежность и масштабируемость программного комплекса.

Все перечисленные параметры следует рассматривать в совокупности. Особое значение имеет возможность интеграции программного продукта, реализующего CRM – функции с уже используемым на предприятии программным обеспечением. Идеальной является ситуация когда CRM – система является частью информационной системы предприятия, тогда вопросы её интеграции с информационной системой предприятия отпадают сами собой.

С этой точки зрения особый интерес представляют собой программные продукты, входящие в систему программ «1С:Предприятие 8». Это связано с тем, что в имеющихся у неё конфигурациях, а именно, «Управление торговлей» и «Управление производственным предприятием» пользователь получает в свое распоряжение набор важнейших CRM – функций как естественное дополнение к её основным возможностям. Немаловажно также и то, что продукты компании «1С» распространяются и внедряются значительным числом её партнеров по всей стране. Практически в каждом регионе России и стран СНГ можно найти непосредственных поставщиков решений компании «1С», готовых оказать услуги по развертыванию и сопровождению системы.

Система программ «1С: Предприятие 8» является универсальной проблемно – ориентированной системой, предназначенной для решения широкого круга задач автоматизации управления предприятием. Она включает в себя единую технологическую платформу и множество конфигураций, определяющих порядок решения задач, специфичных для каждого конкретного предприятия. Особую роль играют так называемые типовые конфигурации, создаваемые фирмой «1С» и включающие наиболее общие универсальные механизмы решения задач организационно – экономического управления предприятием. Как правило, использование системы программ «1С:Предприятие 8» базируется именно на использовании возможностей типовых конфигураций, которые при внедрении на конкретных предприятиях могут быть модернизированы “под специфику” конкретного предприятия.

В соответствии с делением системы на технологическую платформу и прикладные конфигурации, её функционирование разделяется на два относительно независимых режима работы:

1. Пользовательский режим (режим «Предприятие») – режим непосредственного ведения учета организационно–хозяйственной деятельности предприятия.

2. Режим конфигурирования программы (режим «Конфигуратор») – режим администрирования и настройки работы программы, определение состава и структуры объектов программы (констант, документов, справочников и т.д.) и алгоритмов обработки информации, вводимой в систему с их помощью.

С формальных позиций работа с системой состоит в том, что пользователь вводит условно – постоянную и оперативную информацию, относящуюся к хозяйственным операциям предприятия, а система на основе этой информации автоматически производит необходимые расчеты и строит по требованию пользователя необходимые ему отчеты, являющиеся объектами конфигурации. Хозяйственные операции в системе регистрируются путем ввода и проведения документов, также входящих в число объектов конфигурации. Для ввода условно – постоянной информации в систему используются другие объекты программы, а именно, справочники.

В типовых конфигурациях «Управление торговлей» и «Управление производственным предприятием» системы программ «1С:Предприятие 8.*» имеется целый ряд средств и объектов программы, позволяющих развернуть на их основе CRM – систему. Рассмотрим основные из них на примере конфигурации «Управление торговлей 8.0».

Прежде всего, отметим, что в составе главного меню конфигурации, а именно в пунктах «Документы» и «Отчеты» появились отдельные подменю, которые называются «Управления отношениями с клиентами». В

состав этих подменю входят объекты программы, призванные обеспечить технологию CRM в составе конфигурации. Какие это объекты и что они позволяют сделать в плане организации управления взаимоотношениями с клиентами?

Как уже отмечалось, при взаимодействиях с клиентами большое значение имеют контакты с ними. В качестве таких контактов могут выступать как личные встречи, так и обмен электронной или обычной почтой, телефонные звонки и т.д. Менеджерам необходимо помнить обо всех запланированных мероприятиях подобного рода, отслеживать сроки поставок и платежей по сделкам и т.п.

Конфигурация «Управление торговлей» (как и конфигурация «Управление производственным предприятием») располагает возможностями накопления и предоставления такого рода информацией, реализованного за счет механизма событий.

Событие: Личная встреча (исходящая). Проведен *

Личная встреча (исходящая) Действия

Номер: 00000002 от: 16.02.2009 12:00:00 Вид объекта: Контрагент

Общие Дополнительно Участники

Контрагент: ИнноТрейд Начало: 25.02.2009 12:00

Адрес: нет информации Окончание: 25.02.2009 14:30

Контак. лицо: Федоров Борис Михайлович напомнить 25.02.2009 12:00

Телефон: нет информации Источник:

Тема:

Содержание: 1) Обсуждение новой номенклатуры товаров
2) Утверждение сроков оплаты по поставкам

Ответственный: Федоров Борис Михайлович

Комментарий:

OK Записать Закреть

Рис. 2. Документ "Событие"

Этот механизм обеспечивается, в частности, документом «События» (см. рис. 1). В этом документе задается вид события, дата и время начала и окончания, а также краткая информация о его содержании.

События, регистрируемые этим документом, могут быть разных видов (см. рис. 2). Событиям может быть присвоен статус «запланировано», «завершено», «отменено», можно указать на необходимость напоминания о событии в указанный срок.

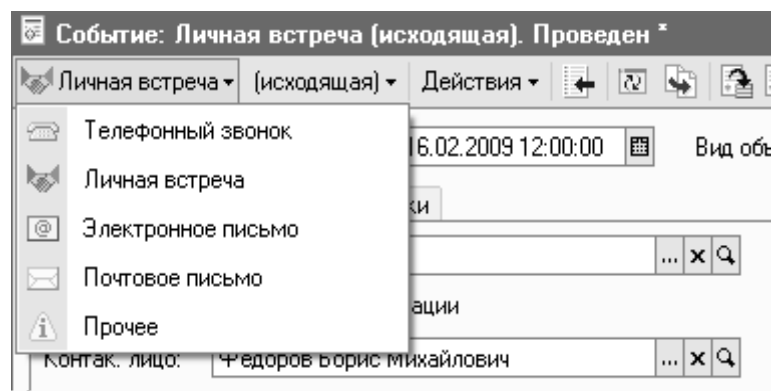


Рис. 3. Виды событий, регистрируемых документом "Событие"

Для удобного представления информации о событиях в программе имеется «Календарь пользователя» в котором помимо сведений о перечисленных ранее событиях, могут быть представлены данные о заказах покупателей и поставщиков и ходе их выполнения (документах оплаты и отгрузки). По каждому заказу прямо из календаря можно формировать отчет «Анализ заказа».

Механизм регистрации и напоминания о событиях, возможности ведения разнообразной справочной информации о клиентах, средства подготовки и исполнения заказов образуют *оперативный блок* CRM – системы, встроенный в систему программ «1С:Предприятие 8».

Коллаборативные функции в программе реализуются за счет общего доступа всех пользователей системы, у которых имеются права, ко всем сведениям о клиентах, имеющимся в информационной базе.

Кроме этого, система «1С:Предприятие 8» обладает развитыми средствами анализа накопленной информации о взаимоотношениях с клиентами, которые образуют *аналитический блок* CRM - системы указанной программы.

Одним из наиболее эффективных инструментов анализа взаимоотношений с клиентами является так называемый ABC – анализ. В его основе лежит принцип Парето, согласно которому 80% оборота (прибыли) компании дают 20% её клиентов [1]. В соответствии с этим принципом всех клиентов фирмы можно разбить на три группы. Как правило, к классу А относятся тех клиентов, которые в совокупности приносят 65-70% выручки. В класс В – выделяют группу клиентов, которая в совокупности приносит 20 - 25% прибыли, а в класс С – всех остальных. Такое деление на группы является достаточно условным, границы групп устанавливает сам пользователь в зависимости от поставленных им целей проведения анализа.

В системе программ «1С:Предприятие 8» основным инструментом проведения ABC – анализа является документ «ABC – классификация покупателей» (см. рис. 3). С его помощью можно автоматически построить

ABC – классификацию клиентов за интересующий пользователя период в соответствии с выбранным параметром ранжирования и заданными границами отнесения их к различным классам.

ABC – классификация дает возможность менеджерам компании и её руководству оценить значимость тех или иных клиентов для её бизнеса и вырабатывать конкретные стратегии взаимоотношений с клиентами из различных групп.

Однако для полноценной борьбы за лояльность клиентов недостаточно знать величину «вклада» того или иного клиента в доходы компании, необходимо также следить за динамикой изменения этих отношений. Если клиент из класса А переходит в класс В то это повод для того чтобы задуматься о причинах этого.

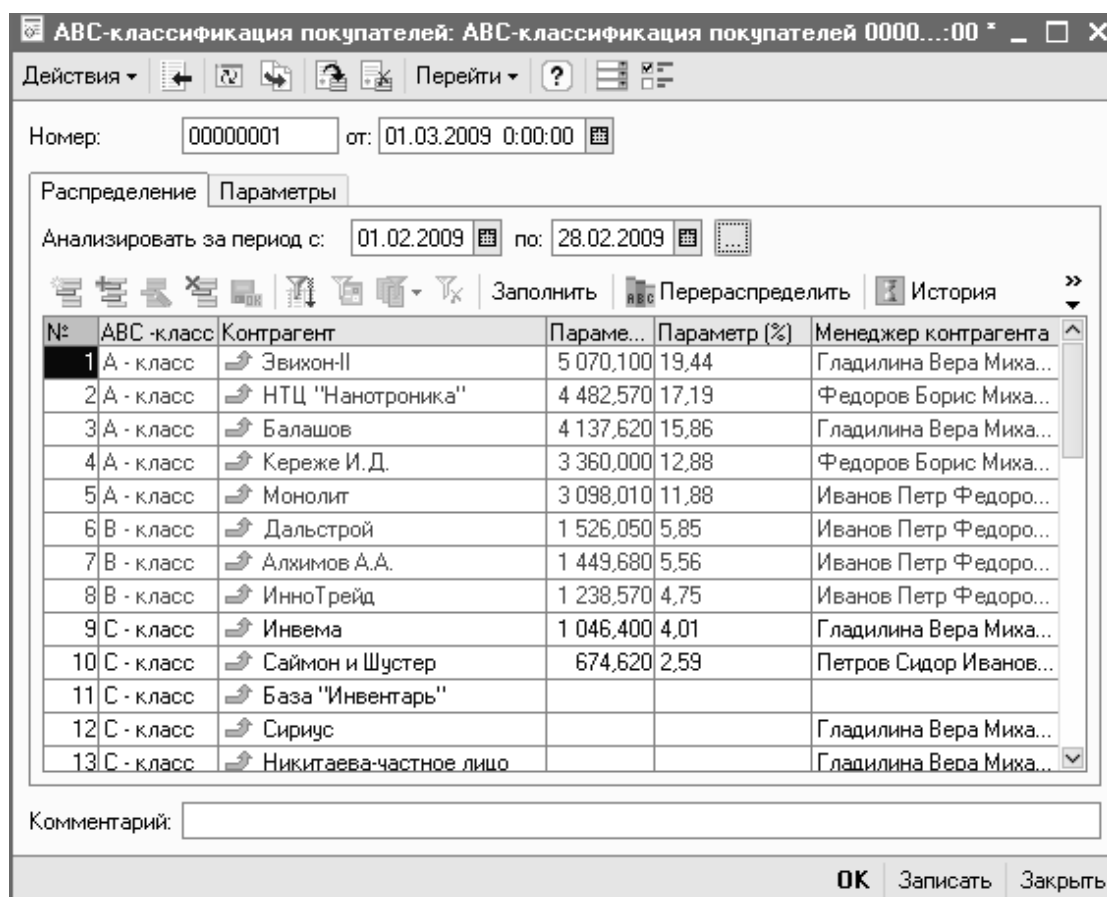


Рис. 3. Документ "ABC - классификация покупателей"

Поэтому слежение за развитием отношений с клиентом целесообразно дополнить анализом стадий взаимоотношений с клиентом. С этой точки зрения клиентов можно разделить на следующие категории:

- потенциальный покупатель;
- разовый покупатель;

- постоянный покупатель;
- потерянный покупатель.

Разделение клиентов на указанные категории, слежение за их «перетеканием» из одной категории в другую, вскрытие причин этого позволяет решать важнейшую задачу для любой компании – прогнозирование объема закупок всеми её клиентами на будущий период.

Для решения этой задачи используется распределение клиентов по стадиям взаимоотношений, в рамках которого проводится так называемый XYZ – анализ. Он предполагает разделение постоянных клиентов на три группы с точки зрения стабильности их закупок. Критерием такого разделения является коэффициент вариации того или иного показателя (выручка от реализации, прибыль и т.д.), характеризующего закупки клиента за ряд последовательных периодов.

В системе программ «1С:Предприятие 8» центральным инструментом рассматриваемого способа классификации клиентов является документ «Классификация покупателя по стадиям взаимоотношений» (см. рис. 4).

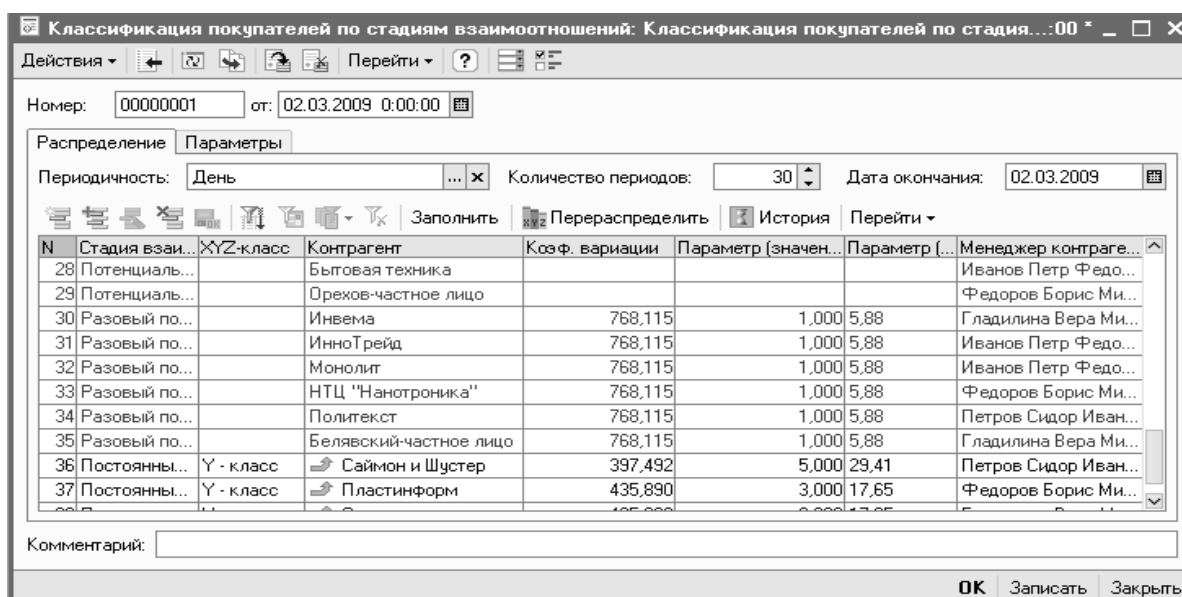


Рис. 4. Документ "Классификация покупателей по стадиям взаимоотношений"

Также как и при проведении ABC – анализа, пользователь может задать критерии отнесения постоянных клиентов к соответствующим классам, а также интервал числа покупок, при котором клиента следует отнести к числу разовых.

Из вышеизложенного следует, что типовые конфигурации «Управление торговлей» и «Управление производственным предприятием» системы программ «1С:Предприятие 8» обладают достаточным набором средств,

позволяющих решать практически все задачи по управлению взаимоотношениями с клиентами, необходимые большинству предприятий.

Кроме того, в заключение отметим, что реализованные в системе «1С:Предприятие 8» CRM – технологии могут быть использованы для анализа взаимоотношений не только с покупателями, но и с поставщиками.

Библиографический список

1. Котлер Ф., Армстронг Г. Основы маркетинга. Профессиональное издание (12-ое издание). - М.: изд-во Вильямс ИД, 2009.- 1072с.
2. Юрьев В.Н. Маркетинговые информационные системы промышленных предприятий. – СПб.: Изд-во СПбГТУ, 1998
3. Карминский А.М., Черников Б.В. Информационные системы в экономике: В 2-х ч. Ч.1 Методология создания: Учеб. Пособие. - М.: Финансы и статистика, 2006.- 336с.

Шубина М.А., кандидат технических наук, доцент

СОВРЕМЕННЫЕ ТРЕБОВАНИЯ К ИСПОЛЬЗОВАНИЮ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ ДЛЯ ЗАДАЧ ЛЕСНОГО ХОЗЯЙСТВА

Современный этап решения лесохозяйственных задач невозможно без широкого использования информационных технологий. В Федеральной целевой программе "Экология и природные ресурсы России (2002 - 2010 годы) один из пунктов сформулирован так: «Научное обеспечение изучения и воспроизводства лесных ресурсов (разработка новых методов, технологий и технологического оборудования для воспроизводства лесных ресурсов, создание информационных ресурсов и систем)».

Известно, что информационные технологии (ИТ) – это совокупность методов и программно-технических средств, объединенных в технологическую цепочку, обеспечивающую сбор, обработку, хранение, распределение и отображение информации с целью снижения трудоемкости процессов использования информационных ресурсов.

Информационные технологии, включающие средства описания информационных объектов; первичного сбора информации; размещения информации в удобном концентрированном виде; хранения информации, управления доступом к информации и извлечения информации, объединяются в геоинформационные технологии; средства для передачи необходимой выборочной информации на расстояние для удобства пользователей, средства для анализа информации и расчета необходимых характеристик – развиваются в рамках других информационных технологий. Но все эти средства базируются на основе современных вычислительных систем, орга-

низация информационных процессов в которых зависит от содержания конкретного набора функциональных задач. Особенно актуальными задачами являются задачи охраны природы и экологической безопасности.

Экология – наука об отношениях растительных и животных организмов и образуемых ими сообществ между собой и с окружающей средой. Объектами экологии могут быть популяции организмов, виды, сообщества, экосистемы, биосфера в целом. Значительное количество статей, конференций, решений Правительств ряда стран свидетельствуют о мировой проблеме предотвращения экологического кризиса – критического состояния окружающей среды, угрожающего существованию человека, вызванного хищническим использованием основных природных ресурсов и загрязнением окружающей среды.

Применительно к экологическим задачам, ИТ включают разработку баз данных и банков данных, обеспечивающих качество, надежность и своевременность получаемой информации и принимаемых решений.

Специфика применения информационных технологий в экологии заключается в том, что в ряде случаев необходимы - специальные датчики для фиксации состояния среды или процессов и, соответственно, средства ввода информации, а также средства для автоматической выработки сигналов предупреждения;

- специальные алгоритмы для анализа получаемой информации;
- средства для моделирования аварийных ситуаций, что приводит к необходимости разработки и внедрения алгоритмов искусственного интеллекта.

В настоящее время по всему миру создано множество организаций, занимающихся экологическим мониторингом и защитой окружающей среды. Достаточно назвать Хельсинскую комиссию; Комитет по природопользованию, охране окружающей среды и обеспечению экологической безопасности; Управление природных ресурсов и охраны окружающей среды и др. Каждый год в марте в Петербурге проводится международный экологический форум «День Балтийского моря». Экологическое сотрудничество администрации СПб с иностранными партнерами берет начало с 1991 г. основополагающими документами являются Хельсинская конвенция (ХЕЛКОМ), Лондонский протокол о воде и здоровье, Конвенция ООН/ЕЭК по защите и использованию трансграничных водотоков и международных озер.

Европейская Комиссия предоставляет информацию по природоохранной политике ЕС и законодательству в ключевых областях по следующим темам: качество воды, управление отходами, качество воздуха, оценка воздействия на окружающую среду, охрана природы, (например, трансграничное сотрудничество) и сети «Natura 2000» (пути организации мер или мониторинга); промышленное загрязнение; которые охватывают следующие сферы: почвы и морская среда (в приоритетной области биоразнообразия),

воздух, пестициды и экология городов (в приоритетной области охраны природы, здравоохранения и качества жизни) и природные ресурсы и переработка отходов (в приоритетной области природных ресурсов и отходов). Природоохранная политика ЕС ставит целью прекратить потерю биоразнообразия к 2010 г. путем:

- восстановления ареалов обитания и естественных экосистем;
- создания сети охраняемых территорий («Natura 2000») и применения соответствующего управления в этих охраняемых территориях.

Разработаны рекомендации о предотвращении загрязнений и таблицы максимально-допустимых концентраций содержания вредных веществ и запрещенных веществ (в первую очередь, ПДК – предельно-допустимых концентраций, МДК – максимально-допустимых концентраций, ФБК - факторов биоконцентрации); соответствующие законы, например, Федеральный закон "Об отходах производства и потребления", Федеральная целевая программа "Экология и природные ресурсы России (2002 - 2010 годы), Приказ МПР "Об утверждении критериев отнесения опасных отходов к классу опасности для окружающей природной среды", СП 2524-82 по сбору, хранению, транспортировке и первичной обработке вторичного сырья и многие другие.

Федеральная целевая программа "Экология и природные ресурсы России (2002 - 2010 годы) включает подпрограммы: "Леса"; "Водные ресурсы и водные объекты", "Регулирование качества окружающей природной среды", "Поддержка особо охраняемых природных территорий", "Сохранение редких и исчезающих видов животных и растений", "Охрана озера Байкал и Байкальской природной территории", "Прогрессивные технологии картографо-геодезического обеспечения" и др.

Учитывая специфику нашего вуза, для нас важными экологическими факторами являются: транспортировка заготовленного леса, особенно водным путем (вспомните сплав леса и загрязнение в связи с этим северных рек), деревообрабатывающие производства, целлюлозно-бумажное производство (ЦБК), производство санитарных и плановых рубок, складирование, своевременный вывоз заготовленного леса, появление вредителей и др.

В качестве показателей экологического состояния могут использоваться как приборы, измеряющие концентрации тех или иных веществ в воздухе, воде, почве, так и состояние флоры и фауны, в том числе, наличие или отсутствие тех или иных микроорганизмов, насекомых или растений.

Для решения проблемы мониторинга окружающей среды активно привлекаются космические методы контроля экосистем. Появление сети Интернет и разработка передовых информационных технологий открыли новый этап развития космического геоэкологического мониторинга. Особенностью нового этапа является широкое использование телекоммуникационной инфраструктуры, гипертекстовых и интерактивных информацион-

ных технологий, которые чрезвычайно перспективны в дистанционном мониторинге состояния окружающей среды. Актуальной является проблема создания региональных баз данных и электронных коллекций космического экологического мониторинга, интегрирования национальных информационных ресурсов и возможность оперативного обмена спутниковыми данными с информационными системами международных космических агентств NASA и ESA.

Сеть Интернет обеспечивает обмен данными дистанционного зондирования Земли и удаленный доступ внешних пользователей к данным и электронным каталогам зарубежных космических программ. Открытый доступ к пространственно распределенным данным долговременных природо-ресурсных и экологических исследований Земли из космоса реализуется на основе Web-технологий. Современной тенденцией развития программ исследования Земли из космоса является создание в ряде стран электронных библиотек космической информации. Эти национальные информационные системы используют потоки спутниковых данных для решения разнообразных задач дистанционного зондирования, определяемых как научным сообществом, так и конкретными отраслями производственной деятельности.

Например, в США для информационной поддержки своей национальной системы наблюдения Земли из космоса NASA создало EOSDIS – разветвленную инфраструктуру сбора, архивирования и распространения спутниковых данных потребителям. Система EOSDIS сосредоточила огромные массивы геопространственных данных, получаемых со спутников ДЗЗ, что создает серьезные проблемы в EOSDIS при организации хранения и доступа к спутниковым данным, поскольку стандартные пакеты программ баз данных не могут их эффективно перерабатывать. Ежедневные объемы необработанных спутниковых данных ДЗЗ, поступающие в систему EOSDIS, оцениваются в 480-490 Гбайт. Объем обработанных данных ДЗЗ в системе EOSDIS достигает 1600 Гбайт в сутки. В настоящее время внешние пользователи через Интернет взаимодействуют с информационной системой наблюдения Земли из космоса EOSDIS. Поисковый сервер EOSDIS базируется на международном стандарте Протокола интероперабельности каталогов (Catalogue Interoperability Protocol – CIP) и позволяет вести поиск одновременно в нескольких каталогах с данными дистанционного зондирования. EOSDIS включает серверы, систему обслуживания словаря данных и путеводитель по метаданным. Информационная система EOSDIS сводит воедино производителей и потребителей спутниковых данных, а также обеспечивает вход в другие информационные системы дистанционного зондирования в Европе (Information on Earth Observation – INFEO) и Японии (CGER).

В России одним из развитых архивов данных дистанционного зондирования является Центр обработки и хранения космической информации ИПЭ РАН (ЦОХКИ ИПЭ РАН), созданный в первую очередь для информа-

ционной поддержки космической программы «Природа». Подробное описание Центра помещено на серверах: <http://www.ire.rssi.ru/priroda> (общие вопросы), <http://www.ire.rssi.ru/cpssi> (центр обработки данных) и <http://www.ire.rssi.ru/cpssi/cpssi.ims.htm> (доступ к данным, включая поиск и заказ данных). ЦОХКИ проектировался, создавался и функционирует в настоящее время в оперативном режиме как полностью интероперабельная система с Электронным Архивом ДДЗ США – системой NASA EOSDIS V0. Поэтому для ЦОХКИ не существует проблемы доступа со стороны пользователей NASA, а также проблем информационного обмена с системой INFEO (Information on Earth Observations), которая представляет собой Научный архив ДЗЗ, созданный Европейской Комиссией в период действия четвертой Рамочной Программы. Анализ запросов по каталогам ЦОХКИ идет в рамках общего опроса (broadcast) по системе EOSDIS. Разработка программного обеспечения спутниковой природо-ресурсной информации направлена на обеспечение в автоматическом режиме решения следующих задач:

Оперативное наблюдение Земли, тематическая обработка ДДЗ и наполнение оболочки Электронного Архива с целью изучения природных ресурсов Земли, геоэкологический мониторинг окружающей среды, оценка состояния экосистем космическими методами, обеспечение доступа к гидрометеорологической информации со всей поверхности Земли.

В процедуре геоэкологического мониторинга территории России одной из существенных проблем, возникающих при наблюдении больших по площади территорий, является необходимость согласования работы нескольких распределенных центров приема и обработки спутниковых данных. Важной проблемой также является организация такой системы доступа к данным, которая была бы максимально независимой от конкретных регионов, где эти данные были приняты и обработаны. Указанные задачи могут быть решены различными способами с помощью сети Интернет.

При существующем в России уровне развития информационных технологий в процедуре спутникового мониторинга все большее значение приобретает принцип распределенной обработки данных. Перспективным оказывается поэтапное создание информационных систем мониторинга, при котором эффективное функционирование информационной системы начинается уже при создании лишь нескольких ее узлов и элементов. Так, первым этапом создания системы доступа к спутниковой информации, принимаемой и обрабатываемой на разных наземных станциях, является разработка базового сервера в узле сети, имеющего развитые телекоммуникации. Информация на центральном (базовом) сервере обновляется различными региональными центрами с использованием даже низкоскоростных коммутируемых каналов. В то же время удаленные пользователи получают доступ по сети Интернет к спутниковым данным, полученным региональными цен-

трами мониторинга. В этой сфере активно работают "Совзонд" и "СканЭкс". В Интернете можно найти изображения Земли в ресурсе "Planeta Earth".

В последнее время информационные ресурсы развитой в России инфраструктуры спутниковой системы геоэкологического мониторинга вызывают значительный интерес зарубежных исследователей. Улучшение доступа к российским космическим данным на основе Web- и Интернет-технологий стимулирует развитие ряда международных программ, объединяющих информационные ресурсы в области мониторинга окружающей среды и уменьшения ущерба от природных и техногенных катастроф. Отметим, что участие в таких программах чрезвычайно важно также для синхронизации и объединения на основе сети Интернет деятельности разрозненных региональных центров приема спутниковых данных и восстановления тем самым потенциала российской спутниковой системы в условиях переходной экономики страны.

Наконец, одним из важнейших инструментов является анализ получаемых результатов, в первую очередь, статистический, и моделирование возможных ситуаций, т.е. алгоритмы искусственного интеллекта.

Для оценки состояния объектов важны комплексные показатели, которые определяются с использованием алгоритмов построения экспертных систем.

Все вышесказанное определяет необходимость улучшения подготовки специалистов, владеющих технологиями комплексной оценки, моделирования ситуаций и дальнейшего их развития и усовершенствования, в том числе, для решения экологических задач за счет расширения курсов по геоинформационным технологиям, (особенно по изучению программных средств, соответствующих решению пункту Программы "Прогрессивные технологии картографо-геодезического обеспечения"), статистической обработке данных и моделированию критических ситуаций.

Библиографический список

1. Федеральная целевая программа "Экология и природные ресурсы России (2002-2010 годы)" от 07.12.2001 №860.
2. Материалы Невского международного экологического конгресса, Спб, март 2008 г. (CDR)
3. http://ec.europa.eu/environment/nature/index_en.htm
4. Аш Е. В., Баринов А.Ю. Геопортал для эффективного решения задач федеральных и региональных органов управления и предприятий. Спб, Семинар ИТЦ СканЭкс, ноябрь 2008 года
5. www.kosmosnimki.ru
6. www.scanex.ru

РАЗРАБОТКА ЭЛЕКТРОННЫХ УЧЕБНЫХ ПОСОБИЙ И ИХ ИСПОЛЬЗОВАНИЕ В ПРЕПОДАВАНИИ ДИСЦИПЛИН КАФЕДРЫ ИНФОРМАТИКИ

Ведущие вузы страны в последние годы уделяют все большее внимание созданию эффективных электронных учебных пособий (ЭУП), иногда называемых электронными образовательными ресурсами (ЭОР), автоматизированными учебными курсами (АУК) или просто электронными учебниками (ЭУ)

Особенно важны такие учебные продукты для систем высшего заочного, дополнительного и дистанционного образования.

Нужны такие пособия и для самостоятельной подготовки студентов в условиях современной образовательной системы, когда на самостоятельную подготовку отводится 50% учебного времени.

Использование ЭУП в образовательном процессе позволяет более углубленно изучить материал, ознакомиться более подробно с интересующими или трудными темами. Необходимый красочно оформленный и мультимедийный иллюстративный материал позволяет наглядно продемонстрировать теоретическую информацию.

Таким образом, студент не только может самостоятельно изучать теоретический материал, останавливаясь на наиболее сложных или интересных непосредственно для него разделах, но и может проработать практические задания по необходимой теме. Студент может выбирать собственную скорость изучения материала.

Материал в ЭУП должен излагаться "прерывисто". Каждый раздел должен завершаться вопросами для самопроверки или тестами, с помощью которых реализуется обратная связь в процессе обучения. Электронный учебник эффективен только при наличии обратной связи; возможности быстрого поиска необходимой справочной информации; демонстрационных примеров и моделей (учебник рассказывает, показывает, объясняет, демонстрирует); когда организован контроль знаний (тренажер, самоконтроль, тестирование). Сочетание этих элементов обеспечивает более быстрое и долговременное усвоение материала электронного учебника.

В отличие от классического варианта учебника, в ЭУП учебный материал представлен в интерактивном изложении, в виде так называемого *гипертекста*, сопровождаемого необходимыми графиками, схемами, рисунками с элементами анимации, а также снабженного как ссылками на различные части данного пособия, так и на другие информационные ресурсы Интернета.

Учебно-информационный текст электронного учебника должен быть четко иерархически структурирован по содержанию. Базовый уровень иерархии отражает основные понятия и концепции предметной области. Бо-

лее высокие уровни должны последовательно детализировать и конкретизировать эти понятия.

Такое пособие, в первую очередь ориентировано на самостоятельное изучение материала студентом, но, безусловно, может быть использовано и на аудиторных занятиях (при наличии соответствующего оборудования для проецирования изображения на экран).

Самостоятельное обучение должно подкрепляться возможностями самодиагностики полученных знаний. Поэтому каждый раздел информационного блока должен заканчиваться контрольными вопросами, которые позволяют студенту выяснить, насколько глубоко он усвоил учебный материал. В результате в ЭУП функционирует постоянная обратная связь обучаемого с «электронным преподавателем», позволяющая повысить эффективность процесса усвоения знаний.

При этом необходимо особое внимание уделить демонстрации такого учебного материала, который является труднодоступным для обучаемого. Речь идет, в первую очередь, о демонстрации процессов с использованием средств мультимедиа и видеотехники в динамическом режиме на конкретных примерах. В курсах информатики и информационных технологий, например, можно организовать просмотр технологии выполнения определенных операций в различных приложениях, демонстрацию выполнения процедур и получение результатов и т.д.

Существуют два вида электронных учебников: закрытый электронный учебник и Internet-учебник.

Первый представляет уже ставший традиционным учебник по предметной области, которой является независимым и неизменяемым. Он используется на персональных компьютерах или в локальных компьютерных сетях и распространяется на дискетах или компакт-дисках.

Под Internet-учебником понимается открытый и имеющий ссылки на внешние источники информации электронный учебник, размещаемый на одном из серверов глобальной компьютерной сети.

Чтобы стать конкурентоспособным специалистом на рынке труда студент должен получить полное представление о современных достижениях в данной предметной области. Для решения этой задачи должна быть разработана такая структура и техническая реализация ЭУП, которая обеспечила бы оперативную модификацию его содержания и дополнение новыми данными, причем эта работа должна выполняться непосредственно составителями учебников без помощи программистов.

В связи с переходом на новые принципы организации обучения, когда на самостоятельное изучение материала студентом отводится 50 процентов времени, отводимого на данный курс, ЭУП позволяют студентам самостоятельно изучать курс.

Специалисты ведущих российских вузов, занимающихся разработкой электронных учебников (Московский государственный институт электрони-

ки и математики (технический университет), Томский Политехнический Университет, Кемеровский государственный университет, РЭА им. Г.В. Плеханова и др.) проведя анализ существующих ЭУП сформулировали основные модули, из которых должен состоять ЭУП, это:

1. Оглавление - содержащее ссылки на все ресурсы (текст, тесты, тренажеры и т.д.) ЭУ.

2. Основные разделы ЭУП, имеющие соответствующие подразделы.

2.1. Теоретический материал. Каждый раздел предваряет структурно-логическая схема, содержащая информацию об основных блоках материала и студент может самостоятельно выбрать тему для изучения.

2.2. Практические задания. Работа в этом режиме дает возможность студенту закрепить знания, полученные при работе с информационным блоком. Система подсказки, при этом, позволяет при необходимости обратиться к любому разделу учебного материала. При неправильном решении необходимо показать студенту правильное решение этой задачи и по его заказу предложить другую аналогичную задачу.

2.3. Тесты или вопросы для самопроверки. В конце каждого теоретического раздела после выполнения практических заданий, должна быть предусмотрена оценка усвоения полученных знаний, в виде тестирования. Тесты могут быть разных типов: выбрать правильный ответ из списка предоставленных, вписать нужное слово и т.д. После выбора ответа студентом, независимо от того, правильный был, выбран ответ или нет, должна быть возможность узнать правильный ответ. Для устранения возможности запоминания студентом правильных ответов необходимо использовать значительный по величине банк тестовых заданий, в котором варианты перемешиваются и берутся в виде произвольной выборки.

2.4. В завершение изучения данного теоретического раздела, выполнения практических заданий и завершения тестирования, студенту должна быть представлена информация о полученных результатах.

3. Предметный указатель - глоссарий или структурированный каталог понятий и модулей, содержащий основные термины со ссылками на определение и местами упоминания их в тексте..

4. Ведение конспекта. При работе с ЭУП студенту предоставляется возможность записывать важную информацию со своими комментариями в виде конспекта, добавлять ссылки на те места в учебнике, где это упоминается. В последствии, он сможет обратиться к своим записям для быстрого поиска нужной информации.

5. Ведение статистики. Система фиксирует время, потраченное студентом на изучение каждой главы учебника, чтобы определить, какому разделу он уделил мало времени или узнать, что ему ещё необходимо прочитать. Также преподаватель сможет определить тему, которую студент знает плохо, а какую вообще не читал.

6. Контроль внимания студента. Программа позволяет определить, как внимательно пользователь работает с текстом. Например, искажается какое-нибудь слово. Если студент внимательно читает текст, то он заметит ошибку, нажмет на это слово, и оно изменится на правильное, а программа зафиксирует это событие. Или другой вариант: через определенный промежуток времени текст учебника будет исчезать, и если студент читает текст, то, переместив мышку, текст вернется, а если студента нет на месте, то время, когда текст был невидим, не будет учтено.

Создание ЭУП включающего все вышеперечисленные разделы достаточно сложная и трудоемкая задача, требующая много времени на полную разработку. Поэтому рекомендуется вводить ее в эксплуатацию по частям:

- в первую очередь разделы 1, 2.1, 2.3 и 3;
- затем разделы 2.2 и 2.4;
- в заключение – разделы с 4-го по 6-й.

Вышеназванная методика подразумевает, что для создания ЭУП, разработчикам необходимы инструментальные оболочки, комплекс программ, по созданию оглавления, предметного указателя, тестов и формы для работы с электронным учебником, которые работают автоматически с минимальным вмешательством человека.

В настоящее время подобные инструментальные оболочки создаются в различных российских вузах, занимающемся развитием системы дистанционного обучения с использованием Интернет-технологий (например, в известном Московском государственном университете экономики, статистики и информатики, МЭСИ - <http://www.ido.ru>). Также созданием подобных инструментальных оболочек-программ занимаются уже несколько лет такие известные **фирмы-разработчики** программного обеспечения как Oracle, Lotus (<http://dlc.miem.edu.ru>), IBM, Maris Multimedia (<http://edugen.com>) и др.

В последние годы работы по созданию электронных учебных пособий ведутся и на кафедре информатики и информационных систем Санкт-Петербургской государственной лесотехнической академии.

Под руководством заведующего кафедрой профессора Заяц А.М., преподаватели кафедры работают над созданием электронного учебного лесничества.

Профессором Заяц А.М. разработаны ЭУП по курсам «Администрирование ИС», «Интернет технологии в ландшафтном проектировании», «Проектирование ИС».

Доцентом кафедры Хабаровым С.П. создано и размещено в Интернете ЭУП (<http://www.habarov.spb.ru>) для самостоятельного изучения курса Информатики, с контрольными вопросами по каждой теме и возможность протестировать полученные знания.

Доцентом Лебедевым М.О. созданы электронные методические указания к лабораторным работам по курсу «Информатика», «Технология про-

граммирования» и «Базы принятия решений» с использованием средств SQL, Visio, Delphi, Access и Excel.

Доцентом кафедры Васильевым Н.П. создан ряд ЭУП для выполнения лабораторных работ и курсового проекта на HTML и JavaScript для студентов лесоинженерного и химико-технологического факультетов.

Доцентом кафедры Ганичевым И.В. разработан комплекс обучающих электронных анимационных моделей – «Стратифицированная модель ЭВМ (иерархическая, 5-уровневая)» и «Цикл выполнения команд ЭВМ».

ЭУП «Основы работы в среде Derive 6» в формате *.chm с использованием «Microsoft HTML Help Workshop» подготовил доцент Джерихов В.В. для проведения лабораторных работ и выполнения курсового проекта студентами ХТБ факультета.

Однако все вышеперечисленные ЭУП предназначены для изучения только дисциплин кафедры ИИС.

Преподавателей кафедры ИИС волнует то обстоятельство, что студенты академии, изучив информатику на 1–2 курсах, в последующем обучении практически не могут применить эти знания при изучении дисциплин других кафедрах.

Созданию подобных учебных пособий для дисциплин других кафедр препятствуют две следующие проблемы.

Во-первых, преподавателям этих кафедр достаточно сложно изучить и работать с инструментальными средствами по созданию таких ЭУП. До 2002 года, пока в ЛТА существовал ИВЦ, в его составе был сектор “Программирования и алгоритмизации задач”, сотрудники которого по заявкам кафедр разрабатывали программы использования компьютеров в учебном процессе этих кафедр. После преобразования ИВЦ в СпецЦНИИТ это подразделение *было* упразднено и внедрение информационно-коммуникативных технологий (ИКТ) в учебные процессы всех кафедр было переложено на сотрудников этих кафедр (в ежегодные министерские отчеты ЛТА должна включать показатели по внедрению ИКТ, и каждая кафедра ежегодно должна передавать эти показатели в УУМО и ТВ ЛТА).

Во-вторых, преподаватели этих кафедр опасаются, что после подготовки таких пособий, студенты получат сокращение времени на решение задач, но будут хуже понимать алгоритм их решения.

Для решения второй проблемы доцентом кафедры ИИС Яриным Б.Д., совместно с доцентом кафедры экономики и управления лесопользования и воспроизводства лесных ресурсов Гавриленко В.И. были разработаны ЭУП для студентов ЛМФ по дисциплинам «Организация производства и менеджмент» и «Эффективность, экономика сервисных услуг и основы предпринимательства». Данное пособие предназначено для выполнения лабораторных работ по решению трех задач с использованием ППП Microsoft EXCEL:

1. «Расчет эффективности использования оборотных средств»;

2. «Определение количества рабочих в бригаде для выполнения текущего ремонта и технического обслуживания лесосечных машин»;

3. «Определение заработной платы ремонтных рабочих в бригаде выполняющей текущий ремонт и техническое обслуживание лесосечных машин».

В данном ЭУП была предпринята попытка не только дать студенту, четкое понятие алгоритма решения вышеуказанных экономических задач, но и показать, как подобные задачи он может решать в своей будущей практической предпринимательской деятельности.

Электронная книга Excel по решению каждой задачи состоит из нескольких листов.

На первом листе приведена постановка задачи с описанием исходных данных и результатов, которые необходимо получить, после решения задачи. Здесь же приведена форма отчета, который студент должен оформить и сдать преподавателю после выполнения работы.

На втором листе студентам напоминаются основные приемы работы в MS Excel, которые они изучали в курсе информатики, но, возможно, забыли.

На третьем - приведены таблицы вариантов студенческих заданий.

На последующих листах (их может быть несколько) подробно рассматривается алгоритм решения данной задачи, приводятся все математические формулы.

Работа студента, после изучения предшествующих листов, начинается с того, что он вводит полученный от преподавателя номер своего варианта, исходные данные по этому варианту автоматически считываются с таблицы вариантов и вводятся в соответствующие ячейки.

Далее, студент сам должен ввести, в отмеченные цветной заливкой ячейки, клеточные формулы, приведенного здесь же алгоритма расчета.

Таким образом, *студент не получает программу решения задачи, а составляет её сам* на языке клеточных формул Excel, тем самым, глубже уясняя алгоритм расчета и получая дополнительную практику в решении задач с помощью ППП Excel.

Все листы книги и ячейки, кроме отмеченных цветной заливкой, защищены от случайной или преднамеренной корректировки.

На последнем листе приведена окончательная форма отчета и сформулированы выводы, которые студент должен сделать после получения результатов расчета.

Опыт выполнения вышеназванных работ показал, что, несмотря на большие трудозатраты, подобные ЭУП можно и нужно создавать не только для дисциплин кафедры ИИС, но и для других экономических и технических кафедр, что такие пособия позволят студентам закрепить знания по специальным дисциплинам и помогут им эффективно решать подобные задачи в своей будущей практической деятельности.

Содержание

<i>Заяц А. М.</i> Итого научно-исследовательской работы	3
<i>Заяц А. М., Савельев Б. В., Гоголевский А. С.</i> Система многопользовательского доступа к ресурсам учебных виртуальных лесничеств	6
<i>Уткин Л. В., Затенко С. И.</i> Интервальная модель тестирования программного обеспечения на основе обобщенных процессов Пуассона	8
<i>Никифоров А. А., Уткин Л. В.</i> Автоматизированная система рейтинговой оценки и мониторинга образовательной и научной деятельности вуза	13
<i>Богатырев В. А., Богатырев С. В.</i> Критерии оптимальности отказоустойчивых серверных систем	17
<i>Богатырев В. А., Осипов А. В.</i> Целочисленная оптимизация многоуровневых компьютерных систем в среде Mathcad	24
<i>Гуров С. В.</i> Оценка надежности сложной системы с зависимыми элементами	30
<i>Терентьев И. В.</i> Оценка погрешности прогнозирования погоды не инструментальными методами	41
<i>Васильев Н. П., Заяц А. М.</i> Опыт использования AJAX технологий	47
<i>Васильев Н. П.</i> Модульность и универсальность – ключи к быстрой разработке web интерфейсов информационных систем	52
<i>Ганичев И. В., Троицкая Е. Г., А.Г. Троицкая.</i> Страты в моделях ЭВМ	56
<i>Ганичев И. В., Васильев Н. П., Киселева С. В.</i> Javascript в информационных системах	58
<i>Ганичев И. В., Троицкая Е. Г., Троицкая А. Г.</i> Анализ и синтез иерархической модели ЭВМ	64
<i>Джерихов В. В.</i> Применение Microsoft Script Editor для разработки и отладки DHTML–страниц	67
<i>Жишко В. Н.</i> Информационный киоск кафедры ИИС ЛТА	73
<i>Курилова М.Н.</i> Цели, задачи и методика проведения олимпиад по информатике	79
<i>Лебедев М. О.</i> Принципы создания динамических журналов для информационно-поисковых и регистрирующих систем	84
<i>Лушкин Н. В.</i> Модель лесохозяйственного производства	88
<i>Лушкин Н. В.</i> Идентификация математической модели процесса растворения в колонном аппарате.....	91
<i>Хабаров С. П.</i> Анализ посещаемости студентами сайта преподавателя в процессе обучения.....	96
<i>Халиков М. И.</i> Информационные технологии ведения взаимоотношений с клиентами в системе «1С: Предприятие 8»	106
<i>Шубина М. А.</i> Современные требования к использованию информационных технологий для задач лесного хозяйства	115
<i>Ярин Б. Д.</i> Разработка электронных учебных пособий и их использование в преподавании дисциплин кафедры информатики	121

Научное издание

ИНФОРМАЦИОННЫЕ СИСТЕМЫ
И ТЕХНОЛОГИИ:
ТЕОРИЯ И ПРАКТИКА

Сборник научных трудов

Выпуск 2

Отпечатано в авторской редакции с готового оригинал-макета

Подписано в печать с оригинал-макета 15.06.09.
Формат 60x84/16. Бумага офсетная. Печать трафаретная.
Уч.-изд. л. 8,5. Печ. л. 8,5. Тираж 100 экз. Заказ № 117. С 217.

Санкт-Петербургская государственная лесотехническая академия
Издательско-полиграфический отдел СПбГЛТА
194021, Санкт-Петербург, Институтский пер., 5