

ФЕДЕРАЛЬНОЕ АГЕНСТВО 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**

С б о р н и к н а у ч н ы х т р у д о в

Выпуск 3
Часть 1

Санкт-Петербург
2010

Рассмотрен и рекомендован к изданию
Ученым советом факультета экономики и управления
Санкт-Петербургской государственной лесотехнической академии
« _____ » _____ 2010 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

С о с т а в и т е л ь

И.В.Ганичев, кандидат технических наук, доцент

Р е ц е н з е н т
доктор технических наук, профессор **В.И.Николаев**
(Санкт-Петербургский филиал ОАО
«Российские телекоммуникационные сети»)

УДК

Информационные системы и технологии: теория и практика: сб.
научн. тр. Вып.3/ от вред. А.М.Заяц. – СПб.: СПбГЛТА, 2010. – 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 2010 года, и практических работ, выполненных ее сотрудниками.

Темплан 2010г. Изд. №217
ISBN 978-5-9239-0163-4
государственная

2010

@ Санкт-Петербургская
лесотехническая академия (СПбГЛТА),

А.М. Заяц кандидат технических наук, профессор

ИТОГИ НАУЧНО – ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЫ

Коллектив кафедры Информатики и информационных систем в 2009 году продолжил проведение исследований ориентированных 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 и управления лесным хозяйством;

2. Применение мультимедийных, Internet\Intranet и WEB - технологий в теоретической и практической подготовке студентов;

3. Разработка научных подходов в методике преподавания учебных дисциплин в рамках магистерской программы «Web - технологии»;

4. Совершенствование методики преподавания учебных дисциплин направления подготовки 230200 «Информационные системы».

По итогам НИР за 2009 год коллектив кафедры занял **15** место из **49** в системе автоматизированной рейтинговой оценки кафедр ЛТА.

В рамках плана мероприятий на кафедре была организована и проведена очередная научно – практическая конференция «*Информационные системы и технологии: теория и практика*», по результатам которой выпущен настоящий сборник трудов, в котором помещены доклады и сообщения преподавателей кафедры и других вузов Санкт- Петербурга.

Преподаватели и аспиранты кафедры принимали активное участие в работе различных конференций и семинаров, на которых выступили с 17- ю докладами.

К наиболее значимым докладам и работам, представленным на российские и зарубежные конференции и форумы следует отнести:

1. Международная научно-практическая конференция «Подготовка научных кадров высшей квалификации в условиях инновационного развития общества», г. Минск

• «Инновационные технологии в практической подготовке и переподготовке управленческих кадров для лесного хозяйства»
А.М. Заяц;

- Опыт внедрения системы автоматизации учебным процессом подготовки и переподготовки управленческих кадров для лесного хозяйства в СПбГЛТА А.М. Заяц, Н.В. Симанова;

2. Третий Санкт-Петербургский конгресс "Профессиональное образование, наука, инновация в XXI веке. «Инновационные образовательные технологии в подготовке специалистов для лесной отрасли» А.М. Заяц;

3. Российско – финляндский лесной саммит. Конкурсная работа - Информационно - образовательный комплекс «Учебные виртуальные лесничества» А.М. Заяц, А.С. Гоголевский

В российских журналах и сборниках трудов ЛТА и других вузов преподавателями опубликовано 15 статей.

В проведении научно-методических исследований активное участие принимали студенты, члены СНО (руководитель доцент Ганичев И.В.) и аспиранты кафедры. Так за отчетный период ими под руководством преподавателей кафедры были представлены 11 работ. Наиболее значимыми из них являются:

1. Труды Санкт Петербургской Государственной Лесотехнической академии. Актуальные проблемы высшей школы «Система многопользовательского доступа к ресурсам учебных виртуальных лесничеств», Б.В. Савельев, А.С. Гоголевский (студенты 5-го курса ЛХФ).

2. Всероссийский смотр – конкурс научно-технического творчества студентов вузов «ЭВРИКА - 2009» Экспертно-консалтинговая информационно-образовательная платформа "Учебные виртуальные лесничества", А.С. Гоголевский.

3. Информационный международный форум «Молодежная волна 2009» Проект – Информационно – образовательный комплекс «Учебные виртуальные лесничества», А.С. Гоголевский.

Продолжается публикация методических материалов и их сопровождение на сайте ЛТА и личных сайтах преподавателей. Так на сайте ЛТА на странице кафедры информатики и информационных систем (<http://ftacademy.ru/academy/faculties/feu/cafedra/iis/>) размещены:

- исторический очерк к 40 – летию кафедры;
- Учебник DHTML от w3school. Автор – составитель

В.В. Джерихов.

На сайте доцента Хабарова С.П. (<http://www.habarov.spb.ru/>) на странице «Информация для студентов» представлены:

- Конспекты лекций учебным дисциплинам в мультимедийной форме: Представление знаний, Экспертные системы, Компьютерные сети, Информатика.

- Лабораторные практикумы и много другой полезной информации для студентов.

На кафедре заканчивается тестовое испытание автоматизированной системы управления деятельностью кафедры, в рамках функционирующей информационной системы (ответственный исполнитель доцент Горбачев В.А.), срок ввода в эксплуатацию которой намечен на 20010 год.

И.В. Панфилов, доктор технических наук, профессор

ОПРЕДЕЛЕНИЕ ТЕХНИЧЕСКОЙ НАДЕЖНОСТИ КОМПЛЕКСА ЭВМ

Требуемую производительность можно обеспечить, объединяя ЭВМ в комплекс. Если каждая из ЭВМ комплекса будет обладать характеристиками, обеспечивающими решение любой отдельной задачи в заданные сроки, то выбором соответствующего числа машин комплекса можно обеспечить требуемую производительность. Наряду с этим необходимо обеспечить требуемую надежность функционирования комплекса.

Под технической надежностью комплекса будем понимать вероятность того, что в комплексе будет работать хотя бы одна из машин в течение заданного времени τ . Однако техническая надежность характеризует только принципиальную возможность решения комплекса всех задач, но без гарантии срока их решения.

Для того, чтобы учесть возможность решения всех задач в установленный срок следует ввести понятие оперативной надежности.

Под оперативной надежностью комплекса понимается вероятность того, что все задачи будут решены в заданный срок. При построении комплекса для обеспечения условия минимума резерва производительности целесообразно сначала обеспечить техническую надежность, а затем обеспечить требуемую оперативную надежность либо добавлением машин в комплекс, либо упрощением алгоритмов, либо оптимизацией программ, либо модернизацией машин, входящих в комплекс, либо созданием и включением в комплекс специализированных машин.

Расчет технической надежности комплекса при выбранном числе n машин можно осуществить следующим образом.

Рассмотрим простейший случай, когда машины, входящие в комплекс, однотипные, потоки отказов и восстановлений каждой машины простейшие с параметрами λ и μ соответственно.

В этом случае вероятности состояний комплекса будут описываться следующей системой дифференциальных уравнений:

$$\left\{ \begin{array}{l} \frac{dP_0}{dt} = -n\lambda P_0 + \mu P_1 \\ \frac{dP_1}{dt} = n\lambda P_0 - [(n-1)\lambda + \mu]P_1 + 2\mu P_2 \\ \dots\dots\dots \\ \frac{dP_k}{dt} = [(n-k+1)\lambda]P_{k-1} - [(n-k)\lambda + k\mu]P_k + (k+1)\mu P_{k+1} \\ \dots\dots\dots \\ \frac{dP_n}{dt} = \lambda P_{n-1} - n\mu P_n \end{array} \right. \quad t=0, \quad P_0=1, \quad P_k=0, \quad k=\overline{1, n} \quad (1)$$

где $P_k(t)$ – вероятность того, что k машин находятся в ремонте в момент t .

Нормирующее условие имеет обычный вид

$$\sum_{k=0}^n P_k = 1$$

Решив систему (1), получим функции $P_0(t), P_1(t) \dots P_k(t)$, выражающие вероятности того, что к моменту времени t будет находиться в ремонте соответствующее число машин.

Если положить при $t=t_0$ $P_0(t_0) = P_0, P_1(t_0) = P_1, \dots, P_n(t_0) = P_n$, то техническую надежность или вероятность того, что с момента $t=t_0$ до момента $t=t_0+\tau$ будет работать хотя бы одна машина, можно определить на основании теоремы о полной вероятности по формуле

$$P(n, \tau) = \sum_{k=0}^{n-1} P_k \cdot P^{(k)}(\tau). \quad (2)$$

где $P^{(k)}(\tau)$ – вероятность того, что на интервале времени $[t_0, t_0 + \tau]$ будет работать хотя бы одна машина при условии, что при $t=t_0$ ремонтировалось « k » машин.

Вероятность P_k можно вычислить из системы при $t \rightarrow \infty$, т.е.

$$P_k = \lim_{t \rightarrow \infty} P_k(t).$$

Так как система стационарна, то из условия

$$\lim_{t \rightarrow \infty} \frac{dP_k(t)}{dt} = 0$$

получим систему алгебраических уравнений для определения P_k .

Решив эту систему при учете нормирующего условия, получим

$$P_k = C_n^k \left(\frac{\lambda}{\mu} \right)^k \cdot P_0 \quad k=0,1,\dots,n-1.$$

$$P_k = C_n^k \left(\frac{\lambda}{\mu + \lambda} \right)^k \cdot \left(\frac{\mu}{\lambda + \mu} \right)^{n-k}$$

Из последних формул легко получим выражение для оценки величины коэффициента готовности комплекса ЭВМ, определяющего вероятность того, что в произвольный момент времени исправна хотя бы одна ЭВМ из комплекса машин

$$K_r^{(n)} = 1 - P_n = 1 - \left(\frac{\lambda}{\lambda + \mu} \right)^n = 1 - \left(\frac{\alpha}{1 + \alpha} \right)^n = 1 - \exp[-nq]$$

где $q = \ln \frac{1 + \alpha}{\alpha}$, $\alpha = \frac{Q}{T}$

Q – периодичность восстановления;

T – периодичность отказа.

Для комплекса, состоящего из одной машины, получаем выражение

$$K_2^{(1)} = \frac{T}{T + Q} = \frac{1}{1 + \alpha}.$$

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

ОПТИМИЗАЦИЯ РАЗБИЕНИЯ КЛАСТЕРА НА ПОДГРУППЫ С ОГРАНИЧЕННОЙ ДОСТУПНОСТЬЮ

Для достижения высокой надежности и отказоустойчивости серверных систем широко используются кластерные архитектуры.

Надежность и эффективность кластерных систем зависит не только от кратности резервирования серверных и коммуникационных узлов, но и от вариантов объединения функционально разнотипных серверов в кластерные группы [2-9].

При наличии n типов серверов по функциональному назначению возможны варианты формирования кластерных групп с объединением:

- однотипных по функциональному назначению серверов [3];
- разнотипных по функциональному назначению серверов всех n типов (полнофункциональные кластерные группы серверов);
- разных по функциональности серверов при неполном функциональном наборе серверов в каждой кластерной группе, при этом выделяются случаи:
 - функциональной непересекаемости кластерных групп;
 - функциональной пересекаемости кластерных групп [4].

Выбор рациональных вариантов объединения серверов в кластерные группы, должен проводиться с учетом минимизации времени обслуживания различных функциональных запросов, максимизации надежности и сглаживания отрицательного влияния отказов серверного и коммуникационного оборудования на снижение функциональности надежности и производительности системы.

Оценка надежности систем при функциональной неоднородности кластерных групп.

Построение рациональных вариантов компьютерных систем кластерной архитектуры требует оценки надежности с учетом вариантов комплектования кластерных групп функционально неоднородными серверами.

В соответствии с [7], будем считать заданными возможные варианты комплектования кластерных групп по функциональному назначению входящих в них серверов, отображаемые матрицей $\|a_{ij}\|$, элемент которой $a_{ij} = 1$, если группа j –го типа комплектации содержит сервер, реализующий i -ю функцию, иначе $a_{ij} = 0$, $j=1,2,\dots,M$. Матрица $\|\varphi_{ij}\|_{n \times m}$, характеризующая функциональные возможности серверных групп системы с учетом их резервирования, формируется из матрицы $\|a_{ij}\|$ с учетом числа (кратности резервирования) групп каждого типа

функциональной комплектации m_1, m_2, \dots, m_M . Структура исследуемых кластерных систем с разделением серверов по группам приведена на рис.1. На рис 1 обозначены коммутаторы верхнего и нижнего уровней (КВУ и КНУ), предназначенные для объединения серверных групп в систему.

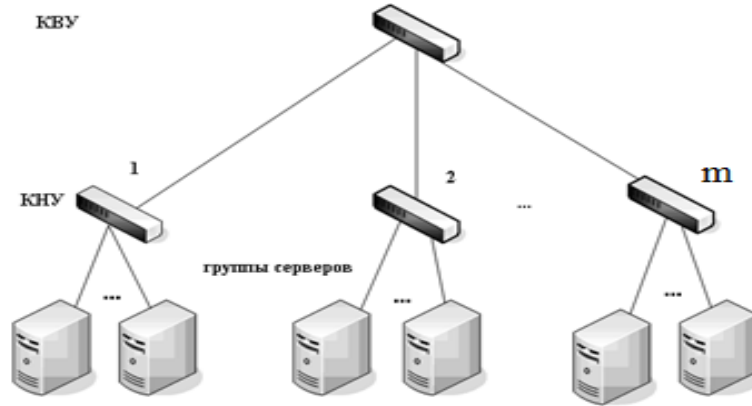


Рис.1. Подключение серверов с сегментацией

В работе [7] предложена оценка надежности систем из многофункциональных модулей. Применяя эту оценку к кластерным системам, в которых (рис. 1) в качестве многофункциональных модулей рассматриваются кластерные группы, включающие разнотипные по функциональности серверы и коммутаторы, вероятность безотказной работы системы вычислим как [8]:

$$P = P_{КВУ} \sum_{k_1=0}^{m_1} \sum_{k_2=0}^{m_2} \dots \sum_{k_M=0}^{m_M} C_{m_1}^{k_1} C_{m_2}^{k_2} \dots C_{m_M}^{k_M} P_{КНУ}^{\sum_{j=1}^M k_j} (1 - P_{КНУ})^{\sum_{j=1}^M m_j - \sum_{j=1}^M k_j} \times$$

$$\times \prod_{i=1}^n (1 - (1 - p_i)^{k_1 a_{i1} + k_2 a_{i2} + \dots + k_M a_{iM}}), \quad (1)$$

где k_1, k_2, \dots, k_M число кластерных групп различной комплектации при исправности входящих в их состав КНУ (возможно резервированных);

$P_{КВУ}$ - вероятность исправности коммутационных узлов верхнего уровня $P_{КВУ} = 1 - (1 - p_0)^{r_0}$, при кратности их резервирования r_0 и вероятности работоспособности одного КВУ равной p_0 ;

$P_{КНУ}$ и p_i вероятности работоспособности совокупности КНУ, и сервера функционального назначения i -го типа. При кратности резервирования КНУ в каждой группе r_1 имеем $P_{КНУ} = 1 - (1 - p_1)^{r_1}$.

Выбор функциональной комплектации кластерных групп

Рассмотрим системы из серверных групп, функциональная комплектация которых представлена матрицами $\|a_{ij}\|$ вида A_1, A_2, A_3, A_4 :

$$\begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}, \quad \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \quad \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \quad \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix},$$

Для примера будем считать кратности резервирования серверных групп, представляемых столбцами матрицы A_1 равными $m_1=8, m_2=8$, матриц A_2, A_3 равными $m_1=8, m_2=m_3=4$, а матрицы A_4 равными $m_1=m_2=m_3=m_4=4$.

Во всех случаях общее число кластерных групп равно 16, а серверов - 32, то есть затраты на реализацию систем одинаковы. Результаты расчета надежности рассматриваемых кластерных систем при $p_i = p_{KHV}$ и $p_i = p_{KHV}^2$ представлены на рис.2. На рис 2 кривая 1 (левая ось $P(p)$) соответствует надежности систем, комплектация серверных групп которых представлена матрицами, S_1, S_4 , а кривые 2, 3 соответствуют - матрицами S_2, S_3 , кривая 4 соответствует системе разнице $DP(p)$ вероятностей работоспособности систем представленных матрицами S_1, S_4 . Расчеты проведены при $p_{KHV} = p, p_1 = p_2 = p_3 = p_4 = p^2$.

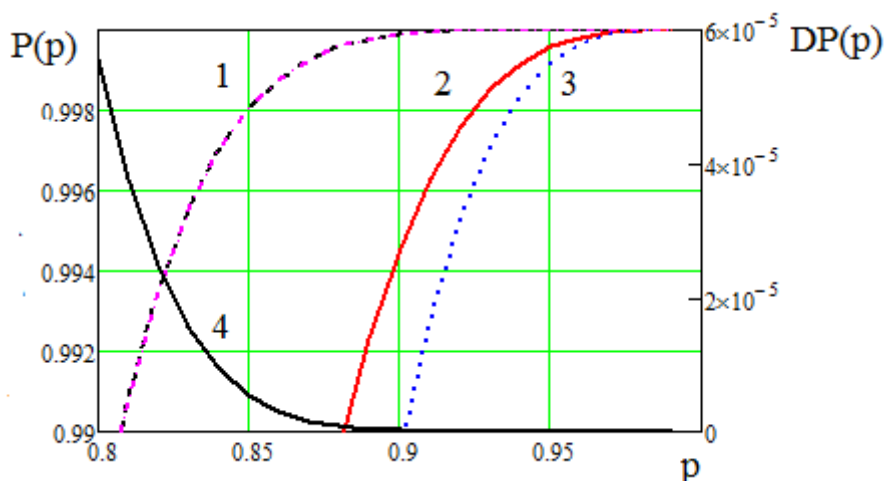


Рис. 2. Надежность системы с различной комплектацией групп

Расчеты подтверждают, что надежность исследуемых систем зависит не только от надежности и кратности резервирования серверов, но и от вариантов их объединения в серверные группы. Расчеты показывают предпочтительность комплектации кластерных групп серверами разного функционального назначения, для которой разница построчных сумм матрицы $\|\varphi_{ij}\|$ минимальна. При выполнении этой рекомендации предпочтительней является распределение серверов по группам, при котором число комбинаций расположения единиц в строках минимально.

Оптимизация кратности резервирования серверных групп

В работе [7] поставлена и решена задача оптимизации для которой при заданном наборе типов многофункциональных модулей (отличающихся функциональной комплектацией), представленным матрицей $\|a_{ij}\|$, требуется определить состав и число m_1, m_2, \dots, m_M модулей каждого типа комплектации, используемых при построении компьютерной системы, обеспечивающих максимум надежности системы при ограничении на стоимость ее реализации C_0 .

$$P(m_1, m_2, m_3, m_4) \rightarrow \max, \quad C(m_1, m_2, m_3, \dots, m_M) \leq C_0$$

В работе [8] задача оптимизации [7] уточнена при рассмотрении в качестве многофункционального модуля группы, включающей серверы и КНУ. При оптимизации надежность системы определяется по (1). Стоимость реализации системы определяется как:

$$C(m_1, m_2, \dots, m_M) = c_0 \sum_{j=1}^M m_j + \sum_{i=1}^n c_i \left(\sum_{j=1}^M a_{ij} m_j \right),$$

Приведем пример оптимизации при стоимости серверных узлов $c_1 = 4, c_2 = 4, c_3 = 4, c_4 = 4$, а стоимость коммутатора $c_0 = 1$, (условных единиц стоимости), если вероятности исправности коммутатора $p_{кн\upsilon} = 0,9$; а серверных узлов $p_1 = 0,9, p_3 = 0,9$; $p_2 = 0,9, p_4 = 0,9$. Зависимости вероятности безотказной работы системы на основе комплектации групп серверов, в соответствии с матрицами A_1, A_4 при ограничении стоимости системы C_0 представлены кривыми 1 и 2 на рис.3. Кривая 3 соответствует разнице надежности для рассматриваемых случаев.

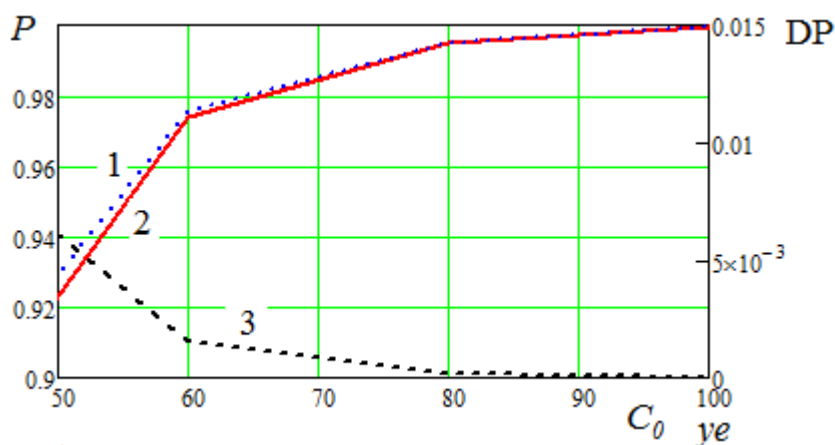


Рис.3. Зависимость надежности системы от ограничений ее стоимости

Возможна другая постановка задачи оптимизации.

Определим состав и число m_1, m_2, \dots, m_M кластерных групп различной функциональной комплектации, обеспечивающих минимум стоимость

реализации системы при обеспечении заданных требований по надежности P_0 .

$$C(m_1, m_2, m_3, \dots, m_M) \rightarrow \min \text{ и } P(m_1, m_2, m_3, m_4) \geq P_0,$$

где c_0 и c_i стоимость коммутатора и i -го типа сервера. Надежность системы определяется по формуле (1), M - число типов комплектации кластерных групп.

Поиск минимума $C(m_1, m_2, \dots, m_M)$ проводится на основе перебора возможных значений (m_1, m_2, \dots, m_M) . При этом задаем начальное значение стоимости системы максимально возможным C_{\max} (например $a=1000$), а затем при переборе (m_1, m_2, \dots, m_M) , если текущее значение стоимости меньше значения переменной «а» и удовлетворяет ограничению по надежности то текущее значение стоимости присваиваем переменной «а» иначе старое значение «а» остается.

Реализация такого алгоритма поиска с использованием средств системы Mathcad-14 приведена ниже:

$$g2(P_0) := \left| \begin{array}{l} a \leftarrow 1000 \\ \text{for } m_1 \in 1..5 \\ \quad \text{for } m_2 \in 1..5 \\ \quad \quad \text{for } m_3 \in 1..5 \\ \quad \quad \quad \text{for } m_4 \in 1..5 \\ \quad \quad \quad \quad a \leftarrow \text{if}(C(m_1, m_2, m_3, m_4, P_0) \leq a, C(m_1, m_2, m_3, m_4, P_0), a) \end{array} \right.$$

При этом

$$C(m_1, m_2, \dots, m_M, P_0) = \begin{cases} \left[c_0 \sum_{j=1}^M m_j + \sum_{i=1}^n c_i \left(\sum_{j=1}^M a_{ij} m_j \right) \right], & \text{if } P(m_1, m_2, \dots, m_M) \geq P_0 \\ C_{\max}, & \text{if } P(m_1, m_2, \dots, m_M) < P_0 \end{cases}$$

В результате оптимизации структуры кластера при выбранной функциональной комплектации кластерных групп можно определить минимум средств и кратность резервирования различных кластерных групп, обеспечивающих заданную надежность кластера.

Приведем пример оптимизации при стоимости узлов $c_1 = 6$, $c_2 = 6$, $c_3 = 2$, $c_4 = 2$, $c_0 = 1$, (условных единиц стоимости) и вероятности их исправности $p_{кпу} = 0,9$; $p_1 = p_3 = 0,7$; $p_2 = p_4 = 0,9$. При заданной предельной надежности системы $P_0 = 0,9995$ и функциональной комплектации кластерных групп в соответствии с матрицей А1 требуется 8 и 10 групп представленных соответственно первым и вторым столбцами, при этом затраты на построение системы равны 154 условным единицам.

Зависимость стоимости реализации системы от требований по ее надежности P_0 при комплектации групп серверов, представляемых матрицами A_1, A_4 представлены кривыми 1 и 2 на рис.4.

Проведенные расчеты (рис. 4) показывают, что требуемый уровень надежности в зависимости от исходной функциональной комплектации кластерных групп может быть достигнут при различных затратах на построение системы.

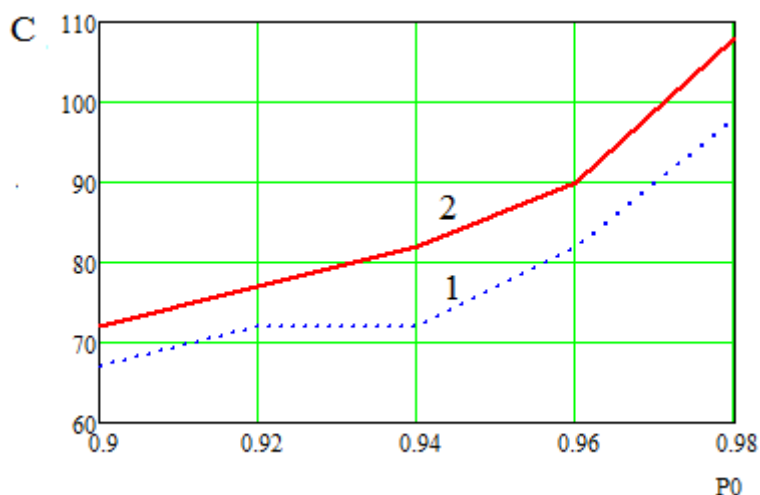


Рис. 4. Зависимость стоимости системы от требований надежности

Полученные результаты могут использоваться при разработке компьютерных систем кластерной архитектуры, в частности, при определении оптимальных по надежности вариантов объединения серверов различного функционального назначения в кластерные группы.

Библиографический список

1. Ретана Ф. Принципы проектирования корпоративных IP-сетей. М.: Вильямс, 2002. 368 с.
2. Богатырев В.А. Оптимизация отказоустойчивых кластеров с неполнодоступностью узлов и неоднородностью потока запросов //Информационные технологии. -2008. № 2. С. 30-36.
3. Богатырев В.А. Оптимальное резервирование системы разнородных серверов //Приборы и системы. Управление, контроль, диагностика. 2007. № 12. С. 30-36.
4. Богатырев В.А., Богатырев С.В. Объединение резервированных серверов в кластеры высоконадежной компьютерной системы //Информационные технологии. -2009. №6.

5. Богатырев В.А. Надежность вариантов размещения функциональных ресурсов в однородных вычислительных сетях // Электронное моделирование. -1997. № 3.

6. Богатырев В.А. Отказоустойчивость распределенных вычислительных систем динамического распределения запросов и размещение функциональных ресурсов. //Наука и образование: электронное научно-техническое издание. 2006. №1. С. 1.0 <http://technomag.edu.ru/doc/56860.html> (<http://elibrary.ru/download/69993264.pdf>)

7. Богатырев В.А. Метод оценки надежности вычислительных систем при функциональной неоднородности компьютерных узлов //Приборы и системы. Управление, контроль, диагностика.- 2006, №12.

8. Богатырев В.А. Оценка надежности и оптимизация комплектации вычислительных систем при резервировании функционально неоднородных компьютерных узлов// Информационные технологии.- 2007, №5.

9. Богатырев В.А., Богатырев С.В. Надежность компьютерных систем при объединении серверов в кластерные группы // Информационные технологии моделирования и управления. 2009. № 4. С. 548-552.

В.А. Богатырев, доктор технических наук, профессор
В.Ф. Беззубов, Е.Ю. Котельникова аспиранты,
А.А. Землянухин, С.И. Белоусов, А.С. Румянцев, А.В. Богатырев
студенты

ВЫБОР СТРУКТУРЫ ОТКАЗОУСТОЙЧИВОГО ДВУХМАШИННОГО ВЫЧИСЛИТЕЛЬНОГО КОМПЛЕКСА

Построение отказоустойчивых систем компьютерного управления базируется на использовании дублированных и троированных вычислительных комплексов [1-5]. Для повышения эффективности распределенного управления вычислительные комплексы объединяются в кластеры. Эффективность кластерных систем определяется структурой объединяемых вычислительных узлов (дублированных комплексов) и организацией взаимосвязи через коммуникационную подсистему, средства которой (магистралей или коммутаторов [6-14]) обычно резервируются. Организация двухмашинных вычислительных комплексов и оценка их эффективности при работе, как в режиме параллельной работы, так и в режиме разделения нагрузки при различных дисциплинах восстановления после отказов известны из [1-5]. В предлагаемой статье рассматривается задача выбора вариантов построения двухмашинных комплексов при

различных подходах к организации их функциональной отказоустойчивости.

При сравнении вариантов построения дублированных комплексов выделены две группы решений по обеспечению отказоустойчивости: в первой предусматривается реконфигурация с переключением ресурсов хранения и обработки между полукомплексами при перераспределении потока запросов; во второй группе реализуется только перераспределение потоков запросов. Восстановление вычислительного процесса на другой машине (полукомплексе) после отказов возможно при периодической организации межмашинного обмена, в том числе для обмена сигналами по таймеру о работоспособности полукомплексов. Межмашинный обмен может проводиться через общую или буферную память по принципу почтового ящика [5], через адаптер межмашинной связи (адаптер канал-канал) или с использованием других известных решений (например, через общую магистраль или коммутаторы локальной сети).

Сравнительный анализ ограничим предположением, что время выполнения запросов в системе не является критичным, и в случае реконфигурации комплекса выполняемый запрос может быть возобновлен сначала без риска срыва процесса управления.

В качестве базового варианта рассмотрим вычислительную машину (ВМ), содержащую процессор и память, связанные между собой через системную магистраль (структура S1- без резервирования). Рассмотрим также конфигурации, включающие две ВМ, при разделении нагрузки (запросов) между ними, при этом выделим варианты, предусматривающие межмашинный обмен через модуль межмашинной связи (адаптер связи – АС модуль межмашинного обмена) и не предусматривающие такой связи. В последнем случае (вариант S2) отказ системы происходит после отказа любого узла одной из двух машин (так как при отсутствии связи невозможно организовать контроль). При объединении ВМ через АС (вариант S3 рис.1) возможна организация контроля работоспособности ВМ, основанного на периодическом обмене информации об исправности машин. В случае не подтверждения работоспособности от другой машины за заданное время ВМ принимает на себя выполнение всего потока запросов (возможно, отбрасывая менее важные задачи).



Рис.1 Конфигурация дублированного вычислительного комплекса с организацией межмашинного обмена

Конфигурации S1 и S2 теряют работоспособность после отказа любого узла, а конфигурация S3 сохраняет функционирование при исправности всех узлов хотя бы в одном полукомплексе.

Таким образом, вероятность работоспособности конфигураций S1-S3 вычисляется соответственно как:

$$\begin{aligned} P_1 &= p_p p_m, \\ P_2 &= (p_p p_m)^2, \\ P_3 &= p_a (1 - (1 - p_p p_m)^2), \end{aligned}$$

где p_p , p_m и p_a - соответственно вероятность работоспособности процессора, блока памяти и адаптера.

При оценке надежности сравниваемых вариантов предположим, что

$$p_p = \exp(-\lambda_p t), p_m = \exp(-\lambda_m t), p_a = \exp(-\lambda_a t).$$

Зависимость вероятности работоспособного состояния исследуемых систем от времени их функционирования t представлена на рис.2. При расчете примем $\lambda_p = 5 \cdot 10^{-5}$ 1/ч., $\lambda_m = 10\lambda_p$.

На рис.2 кривые 1 и 2 соответствуют конфигурациям S1 и S2. Для структур S3 по рис.1, при исправности процессоров обоих полукомплексов, межмашинный обмен возможен как в режиме прямого доступа (ПДП) [14], так и при инициализации обмена по прерыванию, однако, при отказе одного из процессоров в обоих случаях, обращение к памяти полукомплекса с отказавшим процессором становится нереализуемым. Для конфигураций S3, возможна различная реализация адаптеров, от простейших с программным управлением передачей по магистралям двух полукомплексов и обмену через буферную память, до адаптеров, поддерживающих прямой доступ к памяти (ПДП) по двум магистралям одновременно [13], что, как показано в [13], позволяет значительно повысить скорость межмашинного обмена. Для простейшей реализации адаптера примем $\lambda_a = 0,5\lambda_p$ (кривая 3 на рис.2), а для более сложной (кривая 4) предположим $\lambda_a = 0,75\lambda_p$. В качестве промежуточных вариантов могут рассматриваться конфигурации с адаптерами, реализующими программно управляемый обмен по магистрали ВМ инициатора обмена и на основе ПДП по магистрали другой ЭВМ. Учитывая близость кривых 3 и 4 на рис.2 по оси D отражена также их разность кривой 5.

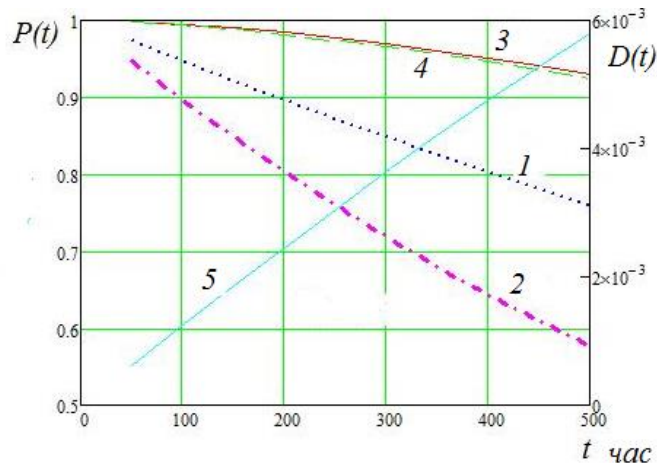


Рис.2. Зависимость вероятности работоспособности сравниваемых систем от времени их функционирования t

Сравниваемые конфигурации характеризуются различной эффективностью обработки запросов. Определим эффективность конфигураций S1 - S3 с учетом вероятностей их работоспособных состояний, характеризуемых различным временем пребывания запросов в системе как:

$$I_1 = p_p p_m \frac{(1-v\lambda)}{v},$$

$$I_2 = (p_p p_m)^2 \frac{(1-\frac{v\lambda}{2})}{v},$$

$$I_3 = p_a (p_p p_m)^2 \frac{(1-\frac{\theta v\lambda}{2})}{v} + 2(1 - (p_p p_m)) p_p p_m \frac{(1-v\lambda)}{v},$$

где v - среднее время выполнения запроса, а λ интенсивность поступления запросов, θ - коэффициент характеризующий замедление выполнения запроса из-за реализации межмашинного обмена для подтверждения работоспособности ВМ комплекса.

Расчет проведем при $v=1$ с. и $t=200$ ч. Результаты расчета эффективности в зависимости от интенсивности запросов λ для конфигурации S1 представлены на рис.3 кривой 1, для S2 кривой 2, а для конфигурации S3 кривыми 3, 4 и 5 при $\theta=1,05$; $\theta=1,1$ и $\theta=1,2$. Из графиков рис.3 видно преимущество конфигурации S3, однако, по мере возрастания интенсивности запросов λ , это преимущество падает.

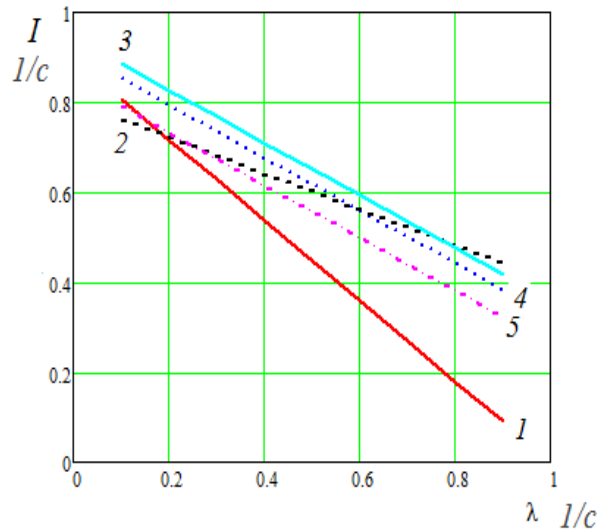


Рис.3. Эффективность сравниваемых конфигураций

Варианты организации S4-S6 дублированных вычислительных комплексов с возможностью перекоммутации (на основе переключателей) ресурсов при реконфигурации (сопровождающейся деградацией) приведены на рис 4.

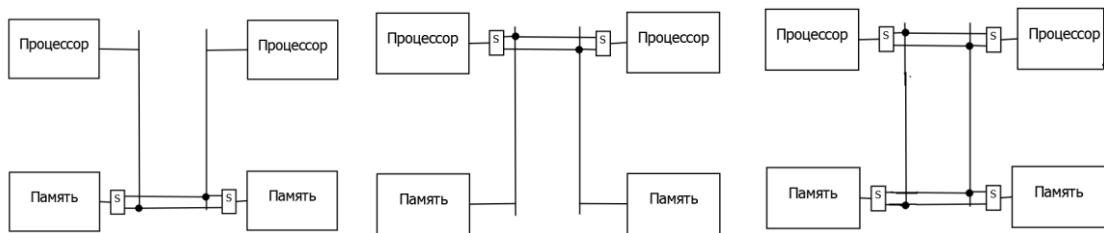


Рис. 4. Варианты организации S4-S6 дублированных вычислительных комплексов с возможностью перекоммутации ресурсов

Для конфигураций S4-S6 будем считать, что переключатели позволяют периодически организовывать доступ к блоку памяти (с организацией «почтового ящика») соседнего полукомплекса, для обмена сообщениями о работоспособности сопряженных полукомплексов. При отсутствии подтверждений работоспособности по таймеру (предположительно после отказа) проводится автоматическое переключение резерва. Вероятность работоспособности конфигураций S4-S6, если переключатель рассматривается в одном из двух состояний (работоспособном и отказавшем), вычисляется как:

$$P_4 = (1 - (1 - p_p p_s)^2)(1 - (1 - p_m)^2),$$

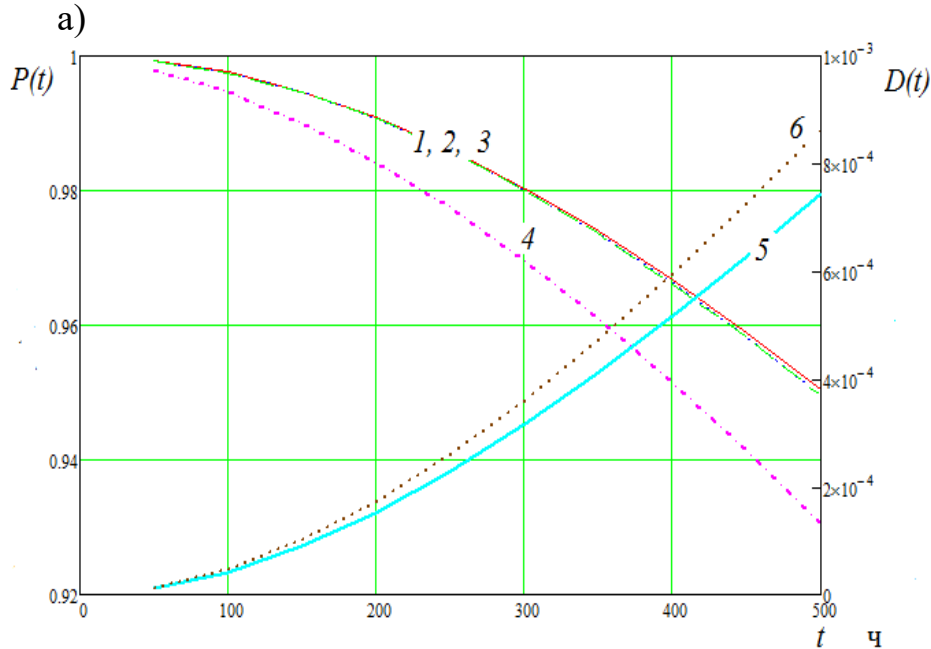
$$P_5 = (1 - (1 - p_m p_s)^2)(1 - (1 - p_p)^2),$$

$$P_6 = (1 - (1 - p_p p_s)^2)(1 - (1 - p_m p_s)^2),$$

где $p_s = \exp(-\lambda_s t)$.

Зависимость вероятности работоспособного состояния конфигураций S4-S6 от времени функционирования t (ч) представлена на рис.5. При расчете примем $\lambda_p = 5 \cdot 10^{-5}$ 1/ч., $\lambda_m = 10\lambda_p$, а интенсивность отказов переключателя будем считать $\lambda_s = 0,1 \lambda_p$ (рис. 5 а) или $\lambda_s = 0,5 \lambda_p$ (рис. 5 б). На рис.5 кривые 1-3 соответствуют конфигурациям S1 - S3, а кривая 4 представляет конфигурацию S3. Учитывая близость кривых 1-3 на рис.5 также представлена разница надежности $D(t)$ для конфигураций S4 и S5 кривой 5 и конфигураций S4 и S6 кривой 6.

Приведенные расчеты показывают явное преимущество построения двух машинных комплексов с обеспечением отказоустойчивости на основе переключения ресурсов (конфигурации S4-S6). Среди конфигураций с переключением ресурсов наихудшей надежностью обладает конфигурация S6, то есть, целесообразно подключение к двум магистралям либо только процессоров, либо только блоков памяти. Подключение через переключатели процессоров, блоков памяти (конфигурация S6) является избыточным и вредным (с точки зрения повышения надежности системы) из-за ненадежности переключателей.



б)

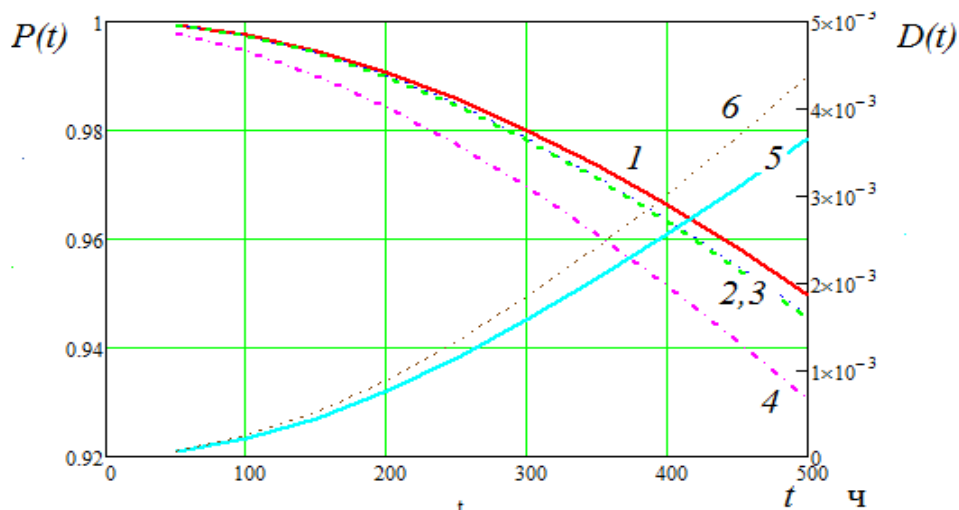


Рис. 5. Зависимость вероятности работоспособного состояния конфигураций S3-S6 от времени функционирования t

Библиографический список

1. Половко А.М. Основы теории надежности. СПб. БХВ-Петербург. 2006 -704 с.
2. Панфилов И.В., Половко А.М. Вычислительные системы. М., «Сов.радио» 1980. 304 с.
3. Вайрадян А.С., Коровин А.В., Удалов В.Н. Эффективность функционирования управляющих мультипроцессорных систем. М., «Сов.радио» 1984. 328 с.
4. Шубинский И.Б., Николаев В.И., Колганов С.К. , Заяц А.М. Активная защита от отказов управляющих модульных вычислительных систем СПб «Наука» 1993. 285 с.
5. Ларионов А.М., Майоров С.А., Новиков Г.И. Вычислительные комплекса системы и сети. Л., «Энергоатомиздат» 1987. 288 с.
6. Богатырев В.А Комбинаторно-вероятностная оценка надежности и отказоустойчивости кластерных систем//Приборы и системы. Управление, контроль, диагностика. 2006. № 6. С. 21-24.
7. Богатырев В.А Оценка надежности и оптимальное резервирование кластерных компьютерных систем //Приборы и системы. Управление, контроль, диагностика. 2006. № 10. С. 18-21
8. Богатырев В.А Оптимальное резервирование системы разнородных серверов // Приборы и системы. Управление, контроль, диагностика. 2007. № 12. С. 30-36.
9. Богатырев В.А Метод оценки надежности вычислительных систем при функциональной неоднородности компьютерных узлов //Приборы и системы. Управление, контроль, диагностика. 2006. № 12. С. 20-22

10. Богатырев В.А Надежность сетей при резервировании коммутаторов с неполнодоступным подключением узлов.//Известия Санкт-Петербургской лесотехнической академии. 2005. № 174. С. 127-134
11. Богатырев В.А., Богатырев С.В. Критерии оптимальности многоустойчивых отказоустойчивых компьютерных систем. //Научно-технический вестник Санкт-Петербургского государственного университета информационных технологий, механики и оптики. 2009. № 5. С. 92-97.
12. Богатырев В.А., Богатырев С.В. Анализ готовности многоуровневой коммуникационной подсистемы компьютерных систем кластерной архитектуры //Научно-технический вестник Санкт-Петербургского государственного университета информационных технологий, механики и оптики. 2009. № 59. С. 88-93.
13. Беззубов В.Ф. Сравнительный анализ методов обмена в многопроцессорных системах. //Вестник компьютерных и информационных технологий. 2006. № 4. С. 51-56.
14. Богатырев В.А., Иванов Л.С., Апинян В.В. Математическая модель мультипроцессорных систем с общей магистралью.//Техника средств связи. Серия Техника проводной связи. 1985 №4. С.113-118.

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

ОЦЕНКА НАДЕЖНОСТИ СЕТИ ZIGBEE

В настоящее время беспроводные сенсорные сети (БСС) находят все более широкое применение в качестве распределенных систем мониторинга различных объектов и физических процессов. Специфика данной задачи определяет общую структуру БСС, которые, как правило, представляют собой распределенные на большой территории информационно-измерительные комплексы.

Наибольшее распространение в последнее время получили БСС, соответствующие стандарту IEEE 802.15.4, а также спецификации профиля протоколов ZigBee [1].

Положение конечных устройств не может выбираться произвольно и определяется конкретными условиями задачи мониторинга, имеющей своей целью контроль требуемых параметров объектов. Ограниченный радиус действия радиопередатчиков, которыми эти узлы оснащены, обуславливает необходимость включения в состав сети ретрансляторов, осуществляющих передачу информационных сообщений от конечных устройств на координатор. Желая минимизировать затраты на создание и

обслуживание беспроводной сенсорной сети, сохранив при этом ее работоспособность и надежность на заданном уровне, мы сталкиваемся с проблемой повышения надежности и отказоустойчивости технического решения.

Одной из заявленных характеристик беспроводных сетей, строящихся на базе стандарта IEEE 802.15.4, является высокая надежность функционирования. Ячеистые сети имеют способность к самовосстановлению. Маршруты могут изменяться в зависимости от изменений радиочастотной обстановки и доступности конкретных узлов. Например, маршрут может изменяться при выходе одного из ретрансляторов из строя или при перемещении конечного устройства. Сеть автоматически перестраивает свою топологию, чтобы обеспечить доставку сообщений узлам.

Для обоснования проектных решений требуется производить количественную оценку параметров сети. Под надежностью в данном случае понимается вероятность безошибочной и своевременной доставки результатов измерений на сетевые шлюзы для дальнейшей обработки.

Надежность БСС определяется многими факторами, наиболее существенными из которых являются: надежность аппаратного и программного обеспечения узлов, область развертывания сети, взаимное расположение узлов, период регламентного обслуживания сети, интенсивность сбора и передачи информации конечными узлами, размер передаваемых пакетов информации.

Подход к оцениванию надежности предполагает использование математической модели надежности передачи данных между двумя узлами БСС, которая, в свою очередь, представляет собой композицию моделей надежности узлов, коммуникаций между ними и механизма их доступа к среде. Модель надежности передачи количественно определяет вероятность успешной передачи пакета.

Для проведения исследования была построена имитационная модель беспроводной сенсорной сети с использованием библиотеки TrueTime [2], функционирующей на базе MATLAB/Simulink. На сегодняшний день существует несколько прикладных решений для имитационного моделирования работы БСС, такие как TOSSIM, который является частью проекта TinyOS, и COOJA, который является частью проекта Contiki. Традиционные сетевые симуляторы, такие как NS-2, также могут быть использованы для симуляции работы БСС. В данной работе для моделирования была выбрана библиотека TrueTime, так как в ней уже имплементирован физический уровень стандарта ZigBee IEEE 802.15.4.

В разработанной модели имитируется работа сети с маршрутизацией по алгоритму AODV (Ad-hoc on-demand Distance Vector routing).

Смоделированная сеть включает семь узлов, размещенных в двумерном пространстве, два из которых изменяют свое положение в процессе симуляции, что вызывает перестроение таблицы маршрутизации. Первоначальное положение узлов приведено на рис. 1.

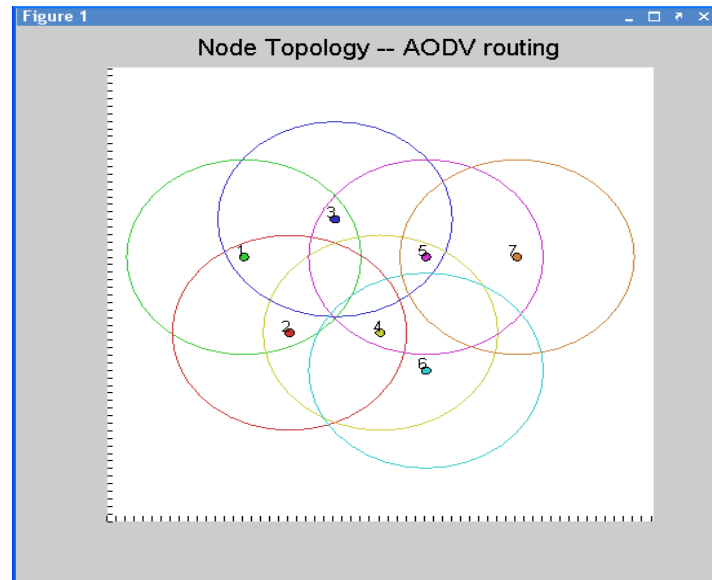


Рис. 1. Изначальное положение узлов сети в пространстве

Надежность передачи пакета данных между двумя узлами может быть выражена вероятностью p_{ij} успешной передачи пакета от i -го узла БСС j -му узлу.

Как известно, в AODV передача маршрутной информации (кроме сообщений Hello) не ведётся, пока нет необходимости в установке или восстановлении маршрута. Когда один из узлов пытается отправить данные в сеть, посылается пакет с запросом на установку маршрута RREQ. Другие узлы сети с AODV пересылают этот пакет в общую среду и делают запись об узле, от которого они приняли запрос, создавая массовую отправку временных маршрутов к запрашивающему узлу. Когда узел получает запрос RREQ и уже имеет маршрут к узлу назначения, то в зависимости от флага «D» сообщения RREQ, он посылает назад сообщение RREP через временный маршрут к узлу требования (узлу-инициатору маршрута) или направляет сообщение RREQ к узлу-получателю, который отправляет сообщение RREP назад узлу-отправителю. Запрашивающий узел в таком случае использует маршрут с наименьшим количеством промежуточных узлов. Неиспользуемые записи в таблицах маршрутизации через некоторое время стираются.

Когда маршрут по каким-либо причинам разрушается, передаётся сообщение об ошибке к передающему узлу и процесс создания маршрута повторяется.

Алгоритм расчета вероятности p_{ij} основан на принципах работы алгоритма AODV и характеристиках, установленных стандартом IEEE 802.15.4 [1]. На каждом i -ом узле имеется ограниченная таблица маршрутизации. Записи в этой таблице ранжированы по предпочтительности использования каждого направления при передаче пакета. Для каждого нового пакета, поступающего на i -й узел и нуждающегося в дальнейшей ретрансляции, делается максимум N_a попыток его передачи j -му узлу, указанному первым в таблице маршрутизации i -го. Если все попытки оказались неудачными, то выбирается $j+1$ узел, соответствующий следующей записи в таблице маршрутизации, и i -й узел пытается передать пакет ему, и т.д. Пакет, который до поступления на i -й узел следующего пакета не удалось передать ни одному из узлов, указанных в таблице, удаляется из системы (теряется).

Ёмкость таблицы маршрутизации, определяющая мощность множества узлов для i -го узла, которым может быть сделана передача, как и максимальное количество повторных попыток передачи N_a не регламентируются стандартами и при построении сети могут быть выбраны произвольно.

В созданной модели БСС согласно сценарию симуляции, самый левый узел (узел №1) периодически посылает информацию узлу №7 каждые 0.5 секунды. В данной модели таблица маршрутизации, созданная по алгоритму AODV, содержит одну строку. Изначально устанавливается следующий маршрут от узла №1 к узлу №7: $1 \rightarrow 3 \rightarrow 5 \rightarrow 7$. В момент времени, равный 3 с., узел №5 начинает движение, что приводит к разрыву этого маршрута. В момент времени (равный 10 с.) узел №6 восстанавливает маршрут, размещаясь между узлами №4 и №7.

Таким образом, выражение для расчета вероятности передачи пакета с первой попытки от i -го узла j -му может быть записано следующим образом:

$$p_{ij} = Q_j \left(1 - \prod_{k=1}^{N_a} (1 - P(k)P_{kw}(\tau_j)) \right),$$

где Q_j - вероятность непередачи i -ым узлом пакета тем узлам, которые находятся выше j -го в его таблице маршрутизации:

$$Q_j = \begin{cases} 1, & j=1 \\ \prod_{z=1}^{j-1} (1-p_{iz}), & j>1 \end{cases}.$$

В созданной имитационной модели таблица маршрутизации содержит всего одну строку, поэтому для вероятности P_{17} передачи пакета с первой попытки от 1-го узла 7-му $Q_7=1$. N_a – максимальное количество неудачных попыток передачи пакета одному узлу; $P(k)$ – вероятность передачи пакета с k -ой попытки: $P(k) = (1-P_1)^{k-1} P_1$, где P_1 – вероятность передачи пакета с первой попытки. Она зависит от таких параметров, как вероятность j -го узла БСС, надежность коммуникации между узлами, определяемая параметрами радиоканала, вероятность коллизии, обусловленной эффектом «скрытого узла».

$P_{kw}(\tau_j)$ – вероятность того, что узел сможет осуществить k попыток передачи пакета за время τ_j . Эта вероятность определяется механизмом конкурентного доступа к среде, используемым в БСС, и регламентированным стандартом IEEE 802.15.4 [1].

Для расчета вероятности успешной передачи пакета могут быть использованы следующие начальные значения:

$$P_1 = 0,75; \quad Q_7 = 1; \quad N_a = 5; \quad P_{kw} = 0,95.$$

Тогда зависимость вероятности успешной передачи пакета от количества использованных попыток примет вид, представленный на рис. 2.

Рассмотренные модели обладают универсальностью и могут найти применение при решении различных задач, связанных с проектированием БСС, например, анализа пропускной способности сети, оптимизации топологии, определении оптимального режима её работы и обслуживания и т.д.

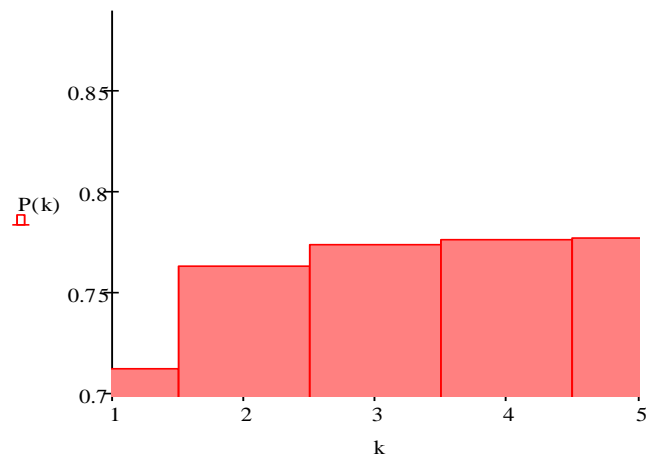


Рис. 2. Зависимость вероятности успешной передачи от использованных попыток

Библиографический список

1. IEEE Standards 802.15.4. Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs). — IEEE Computer Society, 2003.
2. Henriksson, D., A. Cervin, and K.-E. Årzén (2003): “TrueTime: Real-time control system simulation with MATLAB/Simulink.” In Proceedings of the Nordic MATLAB Conference. Copenhagen, Denmark.
3. Ramachandran I., Das A.K., Roy S., «Analysis of the contention access period of IEEE 802.15.4 MAC». — ACM Transactions on Sensor Networks, vol.3, 2007.

С.В., Гуров, доктор технических наук, профессор

НАДЕЖНОСТЬ ТЕХНИЧЕСКОЙ СИСТЕМЫ ПРИ ИЗМЕНЕНИИ УСЛОВИЙ ЕЕ ЭКСПЛУАТАЦИИ

1. Влияние дополнительных воздействий на надежность системы

Функционирование систем часто сопровождается различными внутренними или внешними воздействиями, которые могут влиять на ее надежность. Такие воздействия могут быть вызваны различными условиями эксплуатации системы, изменением нагрузки, функционированием системы в разных климатических зонах, наличием последствий отказов, вмешательством человека в работу системы и др. Указанные воздействия могут носить дискретный или непрерывный характер.

В классической теории под надежностью понимается свойство технического объекта сохранять свои характеристики (параметры) в определенных пределах при данных условиях эксплуатации [1]. Если условия эксплуатации объекта изменяются, то мы вынуждены выйти за рамки классической теории надежности. Ось времени разбивается на промежутки, на каждом из которых может быть свой закон распределения времени до отказа. В зависимости от числа воздействий на систему различных распределений может быть два или большее количество. Возникает задача о том, как связать между собой различные распределения, как осуществить продолжение функции распределения времени до отказа. Это есть задача о стыковке (конкатенации, сцеплении) функций надежности системы на различных этапах ее жизненного цикла.

Предположим, что работа системы на начальном этапе характеризуется вероятностью безотказной работы $P_0(t)$. По каким-либо причинам в момент времени t_1 произошло изменение интенсивности отказов. Например, в момент времени t_1 на систему стала действовать дополнительная нагрузка, которая изменила закон распределения времени до отказа. После момента времени t_1 функция надежности системы стала равной $P_1(t)$. Возникает вопрос о распределении времени до отказа $P_c(t)$ на всем интервале функционирования системы. В работе [2] этот вопрос решался рассмотрением двух систем: без памяти и с памятью, надежность которых выше или ниже надежности исследуемой системы. Тем самым для реальной ситуации определялись двусторонние границы вероятности безотказной работы системы.

Для получения однозначной функции надежности $P_c(t)$ рассмотрим возможные толкования процесса продолжения.

1. Сдвиг графика функции $y = P_1(t)$ по оси y . Тогда $P_c(t) = P_1(t) + x$, где $x = P_0(t_1) - P_1(t_1)$. Этот вид продолжения недопустим, поскольку при $t \rightarrow \infty$ может не выполняться условие $P_c(t) \rightarrow 0$.

2. Сдвиг графика функции ресурса $y = \Lambda_1(t)$ по оси y . Тогда $\Lambda_c(t) = \Lambda_1(t) + x$, где $x = \Lambda_0(t_1) - \Lambda_1(t_1)$. В этом случае для функции надежности имеет место соотношение $P_c(t) = cP_1(t)$, где $c = e^x$. Это значит, что должно произойти искажение графика $P_1(t)$, что, вероятно, также не соответствует действительности.

3. Сдвиг графика функции $y = P_1(t)$ по оси времени t . Тогда $P_c(t) = P_1(t - x)$ при $t \geq t_1$, где x таково, что $P_1(t_1 - x) = P_0(t_1)$. Этот вид продолжения обеспечивает одновременный сдвиг как функции надежности, так и функции ресурса $\Lambda_1(t_1 - x) = \Lambda_0(t_1)$. Он лишен недостатков первых двух способов продолжения.

В дальнейшем будем предполагать, что график функции $P_1(t)$ может перемещаться параллельно самому себе в направлении оси времени, что соответствует возможности его смещения в направлении оси абсцисс (направо или налево). Этот факт можно истолковать как изменение момента времени начала функционирования вспомогательной системы с законом распределения $P_1(t)$. Тогда нужно выбрать такое положение, чтобы график проходил через точку с координатами $(t_1, P_0(t_1))$, обеспечивая тем самым непрерывность функции $P_c(t)$. Приведем графическую иллюстрацию этого свойства.

Пример 1. Пусть $P_0(t) = e^{-0,1t}$, $P_1(t) = e^{-0,2t}$, $t_1 = 7$ ч. Тогда величина сдвига определяется из уравнения (4), т.е. $e^{-0,2(7-x)} = e^{-0,1 \cdot 7}$, откуда $x = 3,5$ ч. Вероятность безотказной работы системы на рис.1 совпадает с кривой $P_1(t)$ при $t \leq 7$ ч., а при $t > 7$ ч. она совпадает с кривой $P_2(t - x)$.

Таким образом,

$$P_c(t) = \begin{cases} P_0(t), & \text{при } t < t_1 \\ P_1(t - x), & \text{при } t \geq t_1 \end{cases}, \quad (1)$$

где x таково, что

$$P_1(t_1 - x) = P_0(t_1). \quad (2)$$

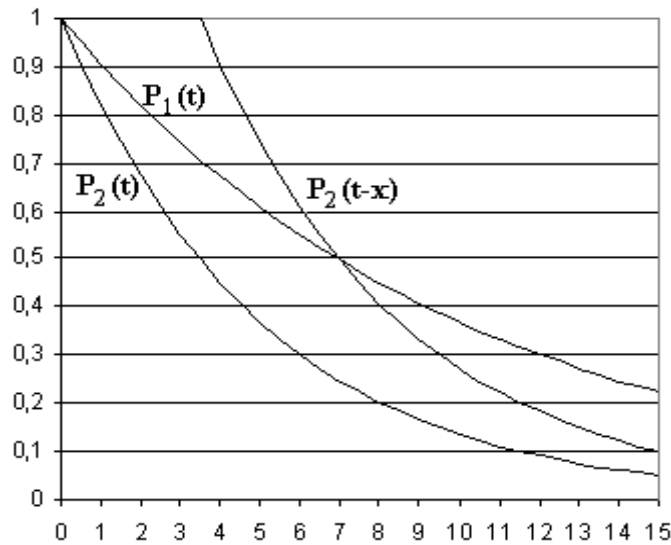


Рис.1.

Пример 2. Начальный этап работы системы характеризуется нормальным распределением времени до отказа с математическим ожиданием $m=10$ ч. и средним квадратическим отклонением $\sigma=2$ ч. Под воздействием некоторых причин в момент времени $t_1=9$ ч. (второй вариант $t_1=11$ ч.) закон распределения времени до отказа изменился и стал экспоненциальным с параметром $\lambda=0,1\text{ч}^{-1}$. Требуется определить результирующий закон распределения времени работы системы с учетом изменившихся условий.

Результирующий закон распределения будем искать по формуле (1), в которой параметр сдвига x вычисляется из соотношения (2). Тогда

$$e^{-\lambda(t_1-x)} = 1 - \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{t_1} e^{-\frac{(t-m)^2}{2\sigma^2}} dt,$$

или, используя обозначение функции Лапласа $\Phi(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^t e^{-\frac{x^2}{2}} dx$, получим

$$e^{-\lambda(t_1-x)} = 1 - \Phi\left(\frac{t_1-m}{\sigma}\right).$$

Отсюда

$$x = t_1 + \frac{1}{\lambda} \ln\left(1 - \Phi\left(\frac{t_1-m}{\sigma}\right)\right).$$

Зависимость функции надежности системы $P_c(t)$ от времени представлена на рис.2 для двух значений $t_1=9$ ч. и $t_1=11$ ч.

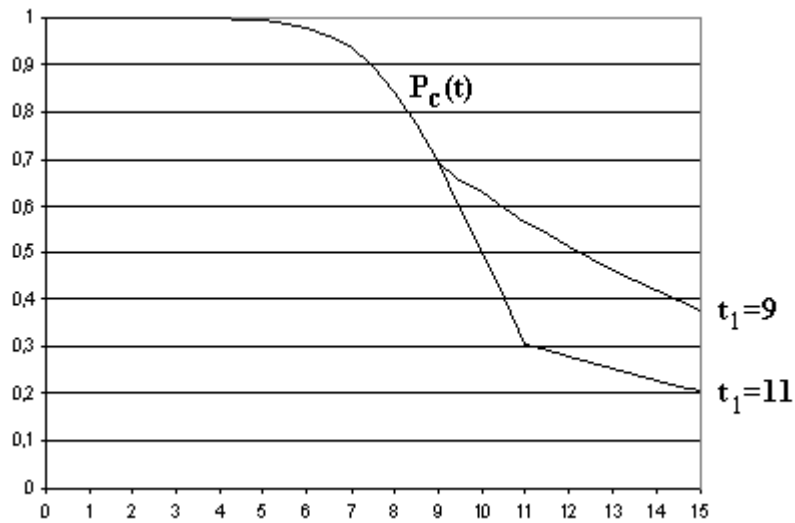


Рис.2.

На рис.2 отчетливо виден переход с кривой Гаусса на график функции экспоненциального распределения. Данный переход соответствует моментам времени 9 и 11ч. Наблюдается также существенная зависимость функции надежности системы от указанных моментов.

2. Функция ресурса системы с переменными условиями эксплуатации

Пусть t_i – моменты изменения закона распределения. На промежутке $[t_i, t_{i+1})$ интенсивность отказа системы равна λ_i , $i=0,1,2,\dots,m$. Здесь $t_0 = 0$, $t_{m+1} = \infty$. Будем говорить, что система с точки зрения надежности характеризуется парой векторов (t, λ) , где $t = \{t_i\}_{i=1}^m$ и $\lambda = \{\lambda_i\}_{i=0}^m$. Обобщая результат стыковки распределений для одного воздействия среды, получим, что вероятность безотказной работы в случае m воздействий для значений $t_i \leq t < t_{i+1}$, равна

$$P_c(t) = e^{-\lambda_i(t-t_i) - \sum_{j=0}^{i-1} \lambda_j(t_{j+1}-t_j)}, \quad i=0,1,\dots,m. \quad (3)$$

Из (6) видно, что при $t_i \leq t < t_{i+1}$ функция ресурса системы выражается равенством

$$\Lambda_c(t) = \lambda_i(t-t_i) + \sum_{j=0}^{i-1} \lambda_j(t_{j+1}-t_j). \quad (4)$$

Если интенсивность отказов системы $\lambda_c(t)$ есть функция времени, то, аппроксимируя ее кусочно-постоянной функцией вида $\tilde{\lambda}_c(t) = \lambda_i$ при $t_i \leq t < t_{i+1}$, получим для функции ресурса соотношение (4). Предельный

переход показывает, что $\Lambda_c(t) = \int_0^t \lambda_c(x) dx$. Таким образом, функция ресурса, рассчитанная по данной методике (с учетом сдвигов функций распределений), совпадает с интегралом от интенсивности. Это есть весомое подтверждение правильности идеи стыковки распределений.

3. Принадлежность функции надежности системы классу $H(r, s)$

Найдем значения r и s , при которых распределение времени до отказа системы принадлежит классу $H(r, s)$. Значение r равно минимальному, для которых функция $\frac{\Lambda_c(t)}{t^r}$ является возрастающей, а s равно максимальному, для которых функция $\frac{\Lambda_c(t)}{t^s}$ – убывающая.

Очевидно $r \leq 1$ и $s \geq 1$, так как при $t < t_1$ функция ресурса имеет вид $\Lambda_c(t) = \lambda_0 t$.

Рассмотрим отношение

$$\frac{\Lambda_c(t)}{t^r} = \frac{\lambda_i(t - t_i) + \sum_{j=0}^{i-1} \lambda_j(t_{j+1} - t_j)}{t^r}$$

на интервале $t_i \leq t < t_{i+1}$.

Для того, чтобы это отношение представляло собой неубывающую функцию необходимо и достаточно, чтобы на интервале $t_i \leq t < t_{i+1}$ производная была бы неотрицательна, иными словами должно выполняться неравенство

$$\lambda_i t^r - \left(\lambda_i(t - t_i) + \sum_{j=0}^{i-1} \lambda_j(t_{j+1} - t_j) \right) r t^{r-1} \geq 0,$$

или

$$(1 - r)\lambda_i t + r\lambda_i t_i - r \sum_{j=0}^{i-1} \lambda_j(t_{j+1} - t_j) \geq 0.$$

Это неравенство верно при всех t из промежутка $t_i \leq t < t_{i+1}$, если оно выполняется при $t = t_i$, т.е.

$$\lambda_i t_i \geq r \sum_{j=0}^{i-1} \lambda_j(t_{j+1} - t_j).$$

Выражение в скобках неотрицательное, поэтому

$$r \leq \frac{\lambda_i t_i}{\sum_{j=0}^{i-1} \lambda_j(t_{j+1} - t_j)}.$$

Таким образом,

$$r = \min \left(1; \frac{\lambda_i t_i}{\sum_{j=0}^{i-1} \lambda_j (t_{j+1} - t_j)}, i = 1, 2, \dots, m \right). \quad (5)$$

Чтобы найти второй параметр класса $H(r, s)$ на интервале $t_i \leq t < t_{i+1}$ рассмотрим отношение

$$\frac{\Lambda_c(t)}{t^s} = \frac{\lambda_i (t - t_i) + \sum_{j=0}^{i-1} \lambda_j (t_{j+1} - t_j)}{t^s}.$$

Это отношение является убывающей функцией, если

$$\lambda_i t^s - \left(\lambda_i (t - t_i) + \sum_{j=0}^{i-1} \lambda_j (t_{j+1} - t_j) \right) s t^{s-1} \leq 0,$$

или

$$(1-s)\lambda_i t + s\lambda_i t_i - s \sum_{j=0}^{i-1} \lambda_j (t_{j+1} - t_j) \leq 0.$$

Поскольку $s \geq 1$, неравенство верно при всех t из промежутка $t_i \leq t < t_{i+1}$, если оно верно при $t = t_i$, т.е.

$$\lambda_i t_i \leq s \sum_{j=0}^{i-1} \lambda_j (t_{j+1} - t_j).$$

Отсюда $s \geq \frac{\lambda_i t_i}{\sum_{j=0}^{i-1} \lambda_j (t_{j+1} - t_j)}$, и, значит,

$$s = \max \left(1; \frac{\lambda_i t_i}{\sum_{j=0}^{i-1} \lambda_j (t_{j+1} - t_j)}, i = 1, 2, \dots, m \right). \quad (6)$$

Соотношения (5) и (6) определяют граничные значения показателей класса $H(r, s)$.

Определим, какому классу распределений принадлежит функция надежности системы из примера 1.

Так как $\lambda_0 = 0,1 \text{ ч}^{-1}$, $\lambda_1 = 0,2 \text{ ч}^{-1}$, $t_1 = 7 \text{ ч.}$, то, как следует из примера 1, функция надежности имеет вид

$$P_c(t) = \begin{cases} e^{-0,1t}, & \text{при } t < t_1 \\ e^{-0,2(t-3,5)}, & \text{при } t \geq t_1 \end{cases}.$$

Согласно (5) получим $r = \min \left(1; \frac{\lambda_1}{\lambda_0} \right) = 1$, а согласно (6) –

$s = \max \left(1; \frac{\lambda_1}{\lambda_0} \right) = 2$. Следовательно, функция надежности $P_c(t)$ принадлежит классу стареющих распределений $H(1, 2)$.

Класс распределений $H(r,s)$ тесно связан с распределениями Вейбулла, поэтому мы сделаем попытку аппроксимации этим распределением вероятность безотказной работы системы с переменными условиями эксплуатации. Распределение Вейбулла, названное в честь шведского исследователя Валоди Вейбулла (Waloddi Weibull), как правило, используется для описания распределения времени безотказной работы механических, электронных и др. устройств.

4. Аппроксимация распределением Вейбулла

Осуществим сглаживание функции надежности системы с изменяющимися условиями эксплуатации распределением Вейбулла на основе равенства первых двух моментов. С этой целью определим среднюю наработку на отказ и дисперсию времени работы системы. Нарботка на отказ:

$$T_1 = \int_0^{\infty} P_c(t) dt = \sum_{i=0}^m \int_{t_i}^{t_{i+1}} e^{-\lambda_i(t-t_i) - \sum_{j=0}^{i-1} \lambda_j(t_{j+1}-t_j)} dt,$$

следовательно,

$$T_1 = \sum_{i=0}^m \frac{1}{\lambda_i} \left(e^{-\sum_{j=0}^{i-1} \lambda_j(t_{j+1}-t_j)} - e^{-\sum_{j=0}^i \lambda_j(t_{j+1}-t_j)} \right). \quad (7)$$

Дисперсия

$$D = 2 \int_0^{\infty} t P_c(t) dt - T_1^2 = 2 \sum_{i=0}^m \int_{t_i}^{t_{i+1}} t e^{-\lambda_i(t-t_i) - \sum_{j=0}^{i-1} \lambda_j(t_{j+1}-t_j)} dt - T_1^2,$$

откуда

$$D = 2 \sum_{i=0}^m \frac{(1 + \lambda_i t_i) e^{-\sum_{j=0}^{i-1} \lambda_j(t_{j+1}-t_j)} - (1 + \lambda_i t_{i+1}) e^{-\sum_{j=0}^i \lambda_j(t_{j+1}-t_j)}}{\lambda_i^2} - T_1^2. \quad (8)$$

Пример 3. Аппроксимировать функцию надежности $P_c(t)$ для данных примера 1 распределением Вейбулла $P_w(t) = e^{-\left(\frac{t}{\beta}\right)^\alpha}$.

В соответствии с (7) и (8) вычислим среднюю наработку на отказ и дисперсию системы: $T_1 = 7,5$ ч., $D = 34,2$ ч².

Параметры распределения Вейбулла рассчитаем на основе известных соотношений [1]:

$$T_1 = \beta \Gamma\left(1 + \frac{1}{\alpha}\right), \quad D = \beta \sqrt{\Gamma\left(1 + \frac{2}{\alpha}\right) - \Gamma^2\left(1 + \frac{1}{\alpha}\right)},$$

из которых получим $\alpha = 1,3$, $\beta = 8,1$ ч.

На рис.3 изображены графики функций $P_c(t)$ и $P_w(t)$. Погрешность аппроксимации – высока.

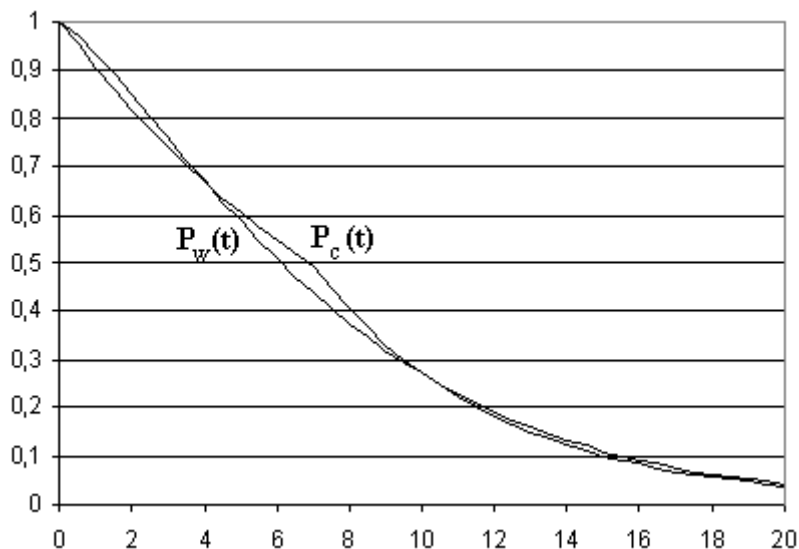


Рис.3. Аппроксимация функции надежности распределением Вейбулла

Далее рассмотрим пример аппроксимации функции надежности системы, подверженной 20 воздействиям различной интенсивности.

Пример 4. В моменты времени t_i в системе происходят количественные изменения интенсивности отказов. На интервалах времени $t_i \leq t < t_{i+1}$ интенсивность отказов постоянна и равна λ_i , $i = 0, 1, 2, \dots, m$. Моменты времени и значения интенсивностей при $m = 20$ приведены в табл.1.

Т а б л и ц а 1

i	t_i	λ_i	i	t_i	λ_i	i	t_i	λ_i
0	0,0	0,100	7	3,5	0,121	14	7,0	0,106
1	0,5	0,100	8	4,0	0,109	15	7,5	0,140
2	1,0	0,152	9	4,5	0,136	16	8,0	0,178
3	1,5	0,110	10	5,0	0,106	17	8,5	0,146
4	2,0	0,142	11	5,5	0,113	18	9,0	0,143
5	2,5	0,142	12	6,0	0,187	19	9,5	0,165
6	3,0	0,156	13	6,5	0,179	20	10,0	0,163

Требуется оценить графически степень близости функции надежности системы с учетом изменения интенсивности отказов и подходящего распределения Вейбулла.

На основе соотношений (7)-(8) определяются средняя наработка на отказ и дисперсия

$$T_1 = 6,6 \text{ ч.}, D = 40,0 \text{ ч}^2.$$

По этим показателям определяются соответствующие параметры распределения Вейбулла

$$\alpha = 1,1, \beta = 7,5 \text{ ч.}$$

По формулам (5) и (6) определяются граничные значения показателей роста функции надежности:

$$r = 0,80, s = 1,51.$$

Значения интенсивностей изменяются не монотонно, поэтому показатель r меньше, а показатель s больше единицы. Графики функций $P_c(t)$ и $P_w(t)$ изображены на рис.4.

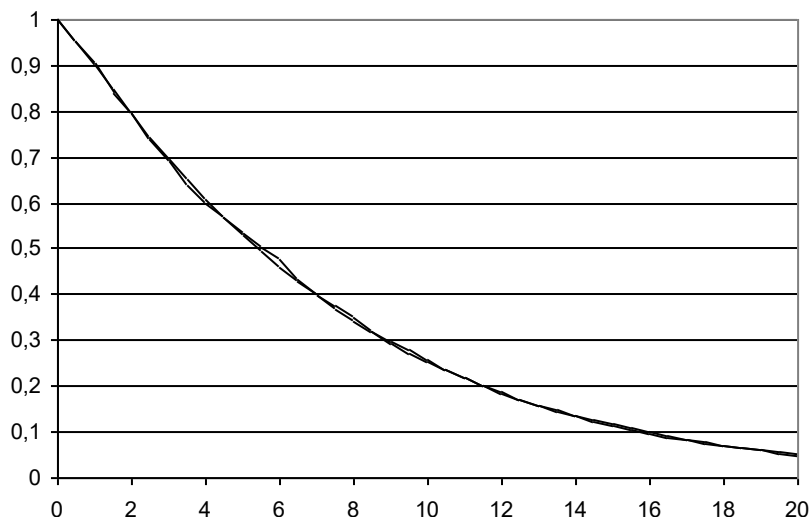


Рис.4. Графики функций $P_c(t)$ и $P_w(t)$ при $m = 20$

Как следует из рис.4, графики функций $P_c(t)$ и $P_w(t)$ при $m = 20$ практически сливаются. Заметим также, что показатель формы α распределения Вейбулла находится в интервале между значениями r и s , а показатель масштаба $\beta = 7,5$ ч. лежит между значениями $\frac{1}{\max \lambda_i} = 5,3$ ч. и

$$\frac{1}{\min \lambda_i} = 10 \text{ч.}$$

5. Сравнение надежности систем при изменении порядка условий эксплуатации

Предположим, что получена функция надежности системы $P_c(t)$ при условии, что до момента времени t_1 вероятность безотказной работы была равна $P_0(t)$, а после момента времени t_1 она стала равной $P_1(t)$. Сделаем допущение, что можно изменить порядок эксплуатации системы на противоположные, а именно, до момента времени t_1 система работает до отказа по закону $P_1(t)$, а после этого момента – по закону $P_0(t)$. Обозначим

через $\bar{P}_c(t)$ – вероятность безотказной работы системы для второго варианта. Спрашивается, какая из систем окажется более надежной.

Получим решение этого вопроса для экспоненциальных распределений с параметрами λ_0 и λ_1 соответственно. Пусть для определенности $\lambda_1 > \lambda_0$. Рассмотрим два случая.

1. До момента t_1 интенсивность отказов системы равна λ_0 , а после момента t_1 она равна λ_1 . Это значит, что после момента времени t_1 система стала менее надежной. Тогда вероятность безотказной работы системы будет равна

$$P_c^{(0,1)}(t) = \begin{cases} e^{-\lambda_0 t}, & t \leq t_1 \\ e^{-\lambda_1 t + (\lambda_1 - \lambda_0)t_1}, & t > t_1 \end{cases}. \quad (9)$$

2. До момента t_1 интенсивность отказов системы равна λ_1 , а после момента t_1 она равна λ_0 . Это значит, что после момента времени t_1 система стала более надежной. Вероятность безотказной работы системы оказывается равной

$$P_c^{(1,0)}(t) = \begin{cases} e^{-\lambda_1 t}, & t \leq t_1 \\ e^{-\lambda_0 t + (\lambda_0 - \lambda_1)t_1}, & t > t_1 \end{cases}. \quad (10)$$

Сравним выражения (9) и (10) графически.

Пример 5. Получить графическое изображение вероятностей безотказной работы системы для двух рассмотренных случаев, если интенсивности отказов соответственно равны $\lambda_0 = 0,01 \text{ ч}^{-1}$ и $\lambda_1 = 0,02 \text{ ч}^{-1}$. Момент смены системой закона распределения времени до отказа t_1 принять равным 20 и 60 час.

Критический момент времени $t_{\text{крит.}}$, соответствующий пересечению функций надежности, определим из уравнения $e^{-\lambda_1 t + (\lambda_1 - \lambda_0)t_1} = e^{-\lambda_0 t + (\lambda_0 - \lambda_1)t_1}$, откуда следует, что он равен $t_{\text{крит.}} = 2t_1$ и нем зависит от надежности системы до и после момента времени t_1 .

Графики функций $P_c^{(0,1)}(t)$ и $P_c^{(1,0)}(t)$ для этих двух моментов показаны на рис.5 и 6.

До момента t_1 более надежной является первая система, имеющая более низкую интенсивность отказа. После момента t_1 до критического момента $2t_1$ первая система по-прежнему является еще более надежной, но после момента $2t_1$ первая система оказывается уже менее надежной по сравнению со второй системой. Из рис.5 и 6 также следует, что с увеличением момента t_1 надежность становится выше для первой системы.

Определим, какая из двух систем более надежная по наработке на отказ.

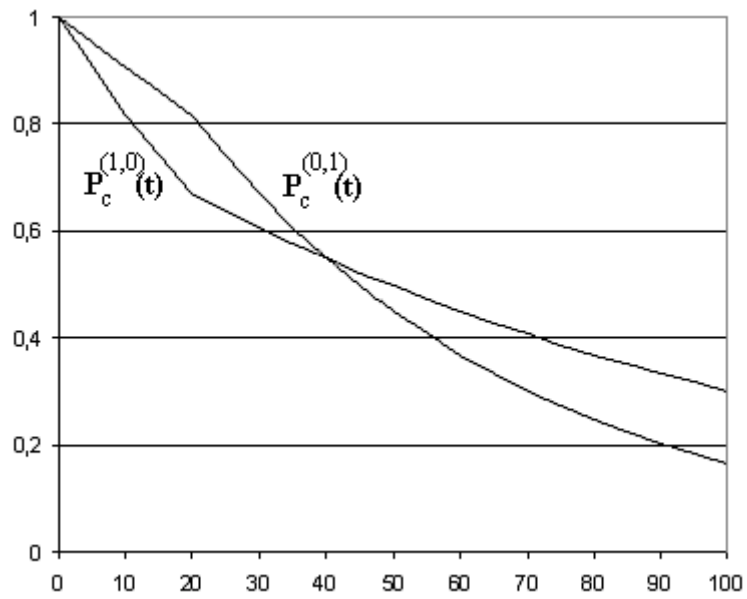


Рис.5. $t_1 = 20$ час

На основании (7) найдем наработку на отказ системы в рассматриваемых случаях

$$T_c^{(0,1)} = \int_0^{\infty} P_c^{(0,1)}(t) dt = \frac{1}{\lambda_0} - \frac{\lambda_1 - \lambda_0}{\lambda_0 \lambda_1} e^{-\lambda_0 t_1}$$

и

$$T_c^{(1,0)} = \int_0^{\infty} P_c^{(1,0)}(t) dt = \frac{1}{\lambda_1} - \frac{\lambda_0 - \lambda_1}{\lambda_0 \lambda_1} e^{-\lambda_1 t_1}.$$

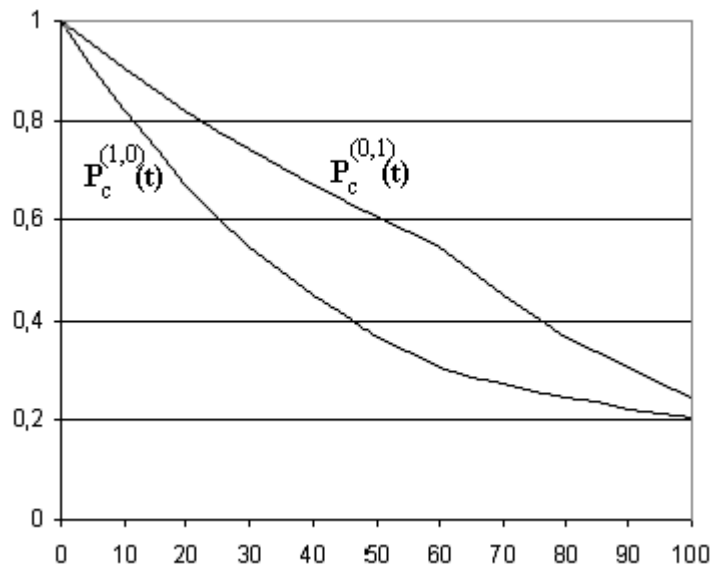


Рис.6. $t_1 = 60$ час

Преобразуем разность наработок.

$$T_c^{(1,0)} - T_c^{(0,1)} = \frac{1}{\lambda_1} - \frac{\lambda_0 - \lambda_1}{\lambda_0 \lambda_1} e^{-\lambda_1 t_1} - \frac{1}{\lambda_0} + \frac{\lambda_1 - \lambda_0}{\lambda_0 \lambda_1} e^{-\lambda_0 t_1},$$

или

$$T_c^{(1,0)} - T_c^{(0,1)} = \frac{\lambda_1 - \lambda_0}{\lambda_0 \lambda_1} (e^{-\lambda_0 t_1} + e^{-\lambda_1 t_1} - 1).$$

Поскольку $\lambda_1 > \lambda_0$, то $T_c^{(1,0)} \geq T_c^{(0,1)}$, если $e^{-\lambda_0 t_1} + e^{-\lambda_1 t_1} \geq 1$

и

$T_c^{(1,0)} < T_c^{(0,1)}$, если $e^{-\lambda_0 t_1} + e^{-\lambda_1 t_1} < 1$.

Таким образом, из условия

$$e^{-\lambda_0 t_1} + e^{-\lambda_1 t_1} = 1 \quad (11)$$

определяется критическое значение момента времени $t_{1,крит.}$, такое, что при $t_1 \leq t_{1,крит.}$ более надежной будет вторая система, у которой после момента времени t_1 условия эксплуатации улучшаются.

При $t_1 > t_{1,крит.}$ более надежной будет первая система, у которой после момента t_1 условия эксплуатации ухудшаются.

Очевидно, что $t_{1,крит.}$ есть функция параметров λ_0 и λ_1 .

Пусть, например, $\lambda_1 = 2\lambda_0$. Тогда из равенства (11) следует, что $e^{-\lambda_0 t_1} + e^{-2\lambda_0 t_1} = 1$, и, значит,

$$t_{1,крит.} = -\frac{1}{\lambda} \ln \frac{\sqrt{5} - 1}{2} \approx \frac{0,4812}{\lambda}.$$

В примере 5 критический момент времени $t_{1,крит.} = 48,12$ ч., и поскольку $t_1 = 7 \leq t_{1,крит.}$, то максимальная наработка на отказ равна $T_c^{(0,1)} = 7,5$ ч. (см. пример 3).

6. Оптимизация режимов эксплуатации системы

Предположим, что система должна пройти несколько фаз, отличающихся условиями эксплуатации. При этом допускается произвольный порядок прохождения фаз. Требуется определить последовательность прохождения фаз, для которой надежность системы будет максимальной.

Интенсивность отказов на каждой фазе различная, она зависит от условий эксплуатации комплекса. Пусть на интервале $t_i \leq t < t_{i+1}$ интенсивность отказов постоянна и равна λ_i , $i = 0, 1, 2, \dots, m$ ($t_0 = 0, t_{m+1} = \infty$).

Конечно, оптимальный порядок будет зависеть от интенсивностей отказов системы на каждой фазе, от моментов смены этих фаз, а также от критерия оптимальности. При $m=1$ данная задача оптимизации решена ранее для двух критериев: вероятности безотказной работы и средней наработки на отказ. Для конкретных значений параметров λ_i и t_i оптимальную последовательность прохождения фаз в общем виде можно

определить путем перебора различных перестановок фаз и выбора из них оптимальной комбинации. Для относительно невысоких значений m (не превышающих 10) эта задача не представляет трудностей для ее решения на компьютере.

Пример 6. Вычислительный комплекс может работать в различных условиях: на суше, в воздухе, на воде с интенсивностями $\lambda_0 = 0,01 \text{ ч}^{-1}$, $\lambda_1 = 0,02 \text{ ч}^{-1}$, $\lambda_2 = 0,03 \text{ ч}^{-1}$ соответственно. Известны времена пребывания комплекса в данных условиях, и, значит, моменты изменений этих условий t_1 и t_2 . Определить оптимальную последовательность фаз, для которой среднее время безотказной работы комплекса будет максимальным.

На основе соотношения (7) получим среднюю наработку, если интенсивности отказов соответствуют комбинации (0,1,2):

$$T_1^{(0,1,2)} = \frac{1}{\lambda_0} (1 - e^{-\lambda_0 t_1}) + \frac{1}{\lambda_1} (e^{-\lambda_0 t_1} - e^{-\lambda_0 t_1 - \lambda_1 (t_2 - t_1)}) + \frac{1}{\lambda_2} (e^{-\lambda_0 t_1 - \lambda_1 (t_2 - t_1)}).$$

Возможно 6 перестановок прохождения фаз и 6 значений наработок, вычисленных в соответствии с порядком их прохождения. В табл.2 приведены наработки для различных моментов t_1 и t_2 (варианты 1,2,3).

Т а б л и ц а 2

Комбинации фаз	Вариант 1	Вариант 2	Вариант 3
	$t_1 = 10 \text{ ч.}, t_2 = 20 \text{ ч.}$	$t_1 = 10 \text{ ч.}, t_2 = 50 \text{ ч.}$	$t_1 = 10 \text{ ч.}, t_2 = 70 \text{ ч.}$
(0,1,2)	42,4	51,6	54,0
(0,2,1)	50,8	51,0	48,5
(1,0,2)	41,5	50,4	56,4
(1,2,0)	76,8	57,0	48,8
(2,0,1)	49,2	49,6	53,3
(2,1,0)	76,0	57,5	52,6
Максимальная наработка, ч.	76,8	57,5	56,4

Для каждого рассмотренного варианта в табл.2 содержится оптимальный порядок прохождения сред, которому отвечает максимальная средняя наработка.

Так, для первого варианта он состоит в работе комплекса в воздухе, на воде, а затем на суше с максимальной наработкой 76,8ч.

Оптимальный порядок для второго варианта состоит в работе комплекса на воде, в воздухе и на суше с максимальной наработкой 57,5ч.

Для третьего варианта оптимальный порядок состоит в работе комплекса в воздухе, на суше и на воде с максимальной наработкой 56,4ч.

Библиографический список

1. Половко А.М., Гуров С.В. Основы теории надежности. Санкт-Петербург, БХВ-Петербург, 2006, 702 с.
2. Половко А.М., Гуров С.В. Основы теории надежности. Практикум. Санкт-Петербург, БХВ-Петербург, 2006, 560 с.

Н.П. Васильев, кандидат технических наук, доцент

СПЛАЙН ИНТЕРПОЛЯЦИЯ С НЕТРАДИЦИОННЫМИ УСЛОВИЯМИ СШИВКИ

Введение. Сплайн интерполяция это тот инструмент, который с успехом используется при решении самых разнообразных задач и прежде всего в компьютерной графике и ее приложениях. Альтернативой является интерполяция по Лагранжу, которая в большинстве приложений, в том числе и в графике, абсолютно непригодна, поскольку склонна к спонтанным осцилляциям, особенно с увеличением количества узлов и, следовательно, с ростом степени интерполяционного полинома. Сплайнам посвящено множество работ, особенно в конце 60-х, начале 70-х годов, и вряд ли можно найти серьезный математический пакет, который обошелся бы без сплайн интерполяции.

Классическая задача интерполяции кубическими сплайнами выглядит следующим образом. Исходными являются узлы интерполяции: $x_i, y_i (i = 0, \dots, n)$. Для каждого из участков x_{i-1}, x_i требуется найти полиномы третьей степени $P_i(x)$, которые удовлетворяли бы классическим условиям сшивки сплайна, а именно:

$$\begin{aligned} P_i(x_{i-1}) &= y_{i-1}, P_i(x_i) = y_i (i = 1, \dots, n) \\ P'_{i-1}(x_i) &= P'_i(x_i) (i = 1, \dots, n-1) \\ P''_{i-1}(x_i) &= P''_i(x_i) (i = 1, \dots, n-1) \end{aligned} \quad (1)$$

Совпадение в узловых точках первых производных обеспечивает отсутствие изломов, а совпадение вторых производных – плавное изменение кривизны результата – составной кривой, проходящей через узлы интерполяции.

Сплайн интерполяция, сохраняющая средние значения. В ряде приложений требуется построить плавную кривую, которая на каждом из участков x_{i-1}, x_i давала бы некоторое заданное среднее значение a_i . Если

вспомнить, как определяется среднее некоторой величины P , изменяющейся на участке x_{i-1}, x_i , то приходим к следующим условиям:

$$\int_{x_{i-1}}^{x_i} P_i(x) dx = a_i h_i, (i = 1, \dots, n) \quad (2)$$

Здесь через h_i обозначена длина отрезка x_{i-1}, x_i . Условия (2) и будут в данном случае играть роль нетрадиционных условий сшивки сплайна. Например, в качестве величины P может выступать температура, для которой известны ее средние значения на определенных участках.

Следует подчеркнуть, что в такой формулировке значения y_i не задаются, как для традиционного случая, а должны быть определены в процессе построения сплайна.

Процедура построения сплайна с сохранением средних значений. Кубический многочлен, представляющий сплайн на i -ом участке, обозначим через $P_i(x)$. Этот кусок составного сплайна должен обеспечивать заданное среднее значение a_i . Воспользуемся следующим представлением куска сплайна, которое можно найти в [1]:

$$P_i(x) = \frac{s_{i-1}(x_i - x)^3 + s_i(x - x_{i-1})^3}{6h_i} + \left(\frac{y_{i-1}}{h_i} - \frac{s_{i-1}h_i}{6}\right)(x_i - x) + \left(\frac{y_i}{h_i} - \frac{s_ih_i}{6}\right)(x - x_{i-1}), (i = 1, \dots, 12) \quad (3)$$

Здесь y_i - значения, которые дает наша искомая кривая в узловых точках x_i , s_i - значения вторых производных в узловых точках. Данное представление удобно тем, что автоматически обеспечивает непрерывность в узловых точках как самого сплайна, так и вторых производных, или иными словами сшивку кусков сплайна по их значениям, а также по кривизне, за которую и отвечает вторая производная. Действительно, можно непосредственно убедиться в том, что

$$\begin{aligned} P_i(x_{i-1}) &= y_{i-1}, P_i(x_i) = y_i (i = 1, \dots, n) \\ P_{i-1}''(x_i) &= s_i, P_i''(x_i) = s_i (i = 1, \dots, n-1) \end{aligned}$$

Таким образом, первое и третье условие сшивки в (1) обеспечиваются автоматически, и остается обеспечить только непрерывность первой производной.

В случае обычной сплайн интерполяции значения y_i в узловых точках x_i задаются, а неизвестные значения вторых производных s_i как раз и определяются из условий сшивки кусков сплайна по первой производной. Наша ситуация является более сложной, поскольку мы не знаем ни вторых производных, ни значений функции y_i в узловых точках.

Взамен неопределенности значений y_i мы имеем соотношения (2). Если в (2) подставить (3), то получим следующие уравнения:

$$y_{i-1} + y_i - \frac{h_i^2}{12}(s_{i-1} + s_i) = 2a_i, (i = 1, \dots, n) \quad (4)$$

Данные уравнения являются дополнительными к традиционным уравнениям сшивки сплайна, обеспечивающим непрерывность первой производной, а, следовательно, и плавность (без изломов) кривой составного сплайна. Традиционные уравнения сшивки сплайна, обеспечивающие непрерывность первой производной, имеют вид [1]:

$$h_i s_{i-1} + 2(h_i + h_{i+1})s_i + h_{i+1}s_{i+1} = 6 \frac{y_{i+1} - y_i}{h_{i+1}} - 6 \frac{y_i - y_{i-1}}{h_i}, (i = 1, \dots, n-1) \quad (5)$$

Таким образом, мы получили систему линейных уравнений (4-5) относительно неизвестных $y_i (i = 0, \dots, n)$ и $s_i (i = 0, \dots, n)$. Очевидно, система не является замкнутой, поскольку неизвестных больше чем уравнений. Действительно, уравнений $2n-1$, а неизвестных $2n+2$.

Дополнительные условия можно получить, если считать концы сплайна свободными, то есть положить $s_0 = 0, s_n = 0$ и из каких-то соображений задаться одним из значений $y_i (i = 0, \dots, n)$. Можно использовать и иные условия на концах сплайна, о которых подробно можно узнать в [1], а также задавать дополнительные соотношения для $y_i (i = 0, \dots, n)$, которые могут диктоваться физикой описываемого процесса.

В результате, получается замкнутая система линейных уравнений, которую можно решить, например, методом исключений Гаусса, и найти неизвестные $y_i (i = 0, \dots, n)$ и $s_i (i = 0, \dots, n)$, которые, в свою очередь, с помощью соотношений (3) позволяют построить составной сплайн. К сожалению, система уже не получается с тремя диагоналями, как для случая классического сплайна, и поэтому нельзя воспользоваться для ее решения более эффективным методом прогонки, позволяющим с большой скоростью строить сплайны для большого количества узлов.

Описанный в данной работе подход был успешно реализован для построения кривой среднегодового хода температур. На рисунке 1 приводится такая кривая. На этом же рисунке построен классический сплайн, проходящий через вершины прямоугольников, представляющих среднемесячные значения температуры – жирная кривая.

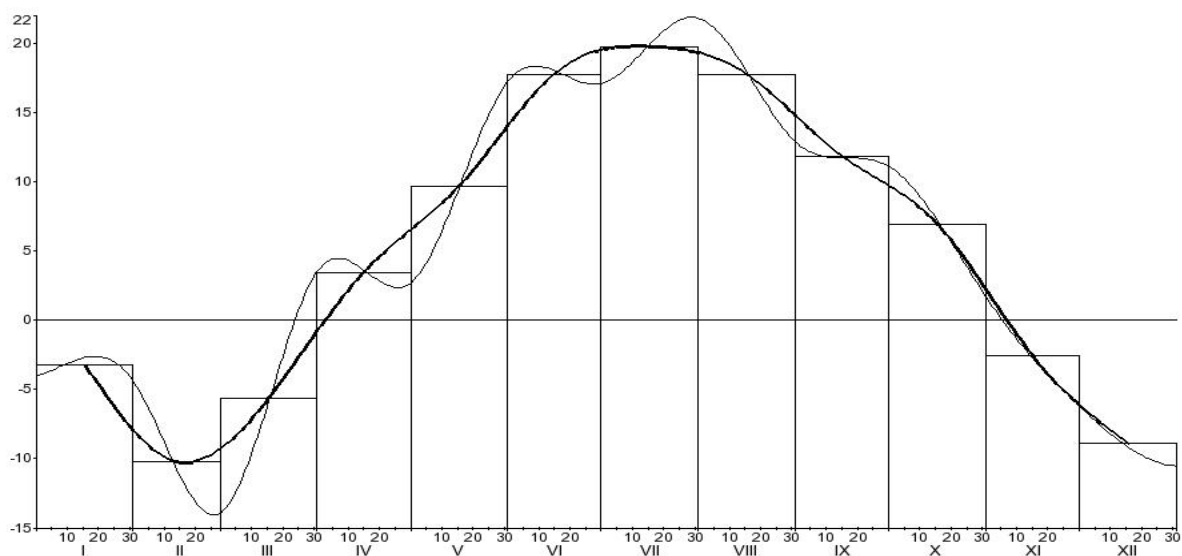


Рис. 1. Годовой ход температуры за 1901 год, полученный двумя различными методами

Библиографический список

1. Фокс А., Пратт М. Вычислительная геометрия. Применение в проектировании и на производстве: Пер. с англ. — М.: Мир, 1982. — 304с.

В.А. Горбачев, кандидат технических наук, доцент

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ФОРМИРОВАНИЯ УСЛОВИЙ ПОИСКА В ЭЛЕКТРОННОМ КАТАЛОГЕ ГЕРБАРИЯ ГЛТА В СРЕДЕ ASP.NET

Задачи и реализация поискового сайта в электронном каталоге гербария. Заполнение базы данными гербария достаточно долгий и непрерывный процесс, потому что необходимо ввести тысячи этикеток гербария. После того как основная масса сведений будет введена в базу, ее можно будет использовать по ее назначению: выполнять поиск сведений о растениях представленных в гербарии. Для этого нужно организовать поиск данных в базе по определенным пользователем критериям. В большинстве систем управления базами данных поиск организуется с помощью запросов. Запрос это формализованное требование на выборку данных из базы. Формирование запроса состоит из следующих этапов:

- формулирование вопроса пользователем базы данных;
- определение списка полей базы данных, значения которых необходимо посмотреть;
- определение таблиц, в которых содержатся выбранные поля;

- установление связей между таблицами;
- формирование условий отбора по значениям выбранных полей.

Формулирование вопроса выполняется пользователем. Все возможные вопросы к базе данных можно сгруппировать по направлениям:

- поиск экземпляров коллекции по элементам систематики: семействам, родам, видам;
- поиск экземпляров коллекции по географии сбора, флористическим и геоботаническим районам;
- поиск по авторским атрибутам: имени коллектора, определителя, по именованным коллекциям;
- поиск по местообитанию: элементам ландшафта и типам растительности.

В большинстве случаев поля, относящиеся к перечисленным группам, могут перемешиваться.

Задача создания поисковой системы электронного каталога гербария заключается в разработке WEB формы, в которой представлены группы условий поиска и результаты поиска, представленные содержимым карточек коллекции, отвечающих выбранным признакам поиска. Пользователь выбирает значения признаков поиска и получает список карточек коллекции. Далее условия могут быть уточнены или изменены и поиск повторен.

Общий вид WEB формы реализующей указанные задачи имеет следующий вид.

Рис. 1. WEB форма поиска данных в электронном каталоге гербария им. Бородина

Нужно отметить, что это первоначальный вид формы. В дальнейшем дизайн предполагается изменить. Форма примет более привлекательный для пользователя вид и в нее будут добавлены изображения и описания типовых образцов.

Рис. 2. Блок формирования условий поиска в гербарии

Как видно из фрагмента экрана в условия поиска включены сведения о систематике, географии сбора, принадлежности к коллекции, автора сбора и времени сбора. Включение в список условий поиска всех данных карточки не целесообразно по следующим соображениям. Блок коллекции не включен в условие поиска, так как это внутренняя, не общепринятая градация коллекции. Область в дальнейшем будет добавлена в условие поиска, если будет востребована по результатам опытной эксплуатации системы. включена в условие поиска в виде года сбора экземпляра коллекции. Указания дня и месяца сбора не целесообразно, так как во многих экземплярах карточек коллекции заданы месяц и год, или только год. Результаты эксплуатации WEB формы покажут, будет ли необходимость делать поиск по природно-климатическим зонам, типу растительности, флористическим и геоботаническим районам и элементам ландшафта. Условие поиска составляется из набора значений представленных характеристик гербария, которые могут быть выбраны в произвольном порядке. Например, выбирается семейство и наименование коллекции, где оно встречается. Далее нажимается кнопка «ПОИСК» и происходит выборка экземпляров карточек из электронного каталога. Результат выводится на экран, как показано на рис. 3.

Рис. 3. Результат поиска в электронном каталоге гербария

Одновременно на экране видно содержимое одной карточки. Переход по карточкам осуществляется с помощью указателей экземпляров ответов обозначенных цифрами в нижней строчке.

Продолжение поиска может идти по пути уточнения условий поиска путем выбора дополнительных характеристик поиска и нажатием кнопки «ПОИСК». Или же нажимается кнопка «СБРОС», в результате чего все характеристики поиска устанавливаются в исходное значение и условия поиска формируются заново.

Каждая из характеристик экземпляра коллекции в условии поиска в гербарии выбирается из списка, который формируется из карточек коллекции. Поэтому при выборе характеристик входящих в условие необходима настройка каждой из них в зависимости от ранее выбранных. Например, пользователь хочет посмотреть экземпляры карточек коллекции семейства XXXXXXXX. Для этого он выбирает в поле «Семейства» соответствующее название. Если, далее, он захочет выбрать коллектора, то в списке коллекторов должны появиться только те коллекторы, которые имеют отношение к сбору экземпляров гербария данного семейства. Если в списке оставить всех коллекторов, то выбор коллектора, не имеющего отношения к данному семейству даст пустой результат поиска. Это вызывает критические замечания пользователей.

Программная реализация WEB формы поиска в электронном каталоге гербария

Описываемая WEB форма реализована с помощью ASP.NET поддерживаемой Visual Studio 2008. Создаваемый в результате проектирования WEB формы код состоит из декларативного описания всех управляющих элементов WEB формы и программной составляющей, обрабатывающей события формы. Опишем основные принципиальные моменты проектирования. Помимо операторов HTML описывающих структуру, стили и размещение элементов управления документа, коде документа присутствуют специфические для ASP.NET элементы управления - контролы, позволяющие связывать элементы управления формы с базой данных.

Поскольку база данных электронного каталога гербария функционирует под управлением SQL Server 2005, то таким контролом является SqlDataSource. Принцип его использования рассмотрим на примере работы поля со списком (DropDownList) «Семейства». Описание элемента в WEB форме выглядит следующим образом.

```
<asp:DropDownList ID="Семейства" runat="server"  
    DataSourceID="SqlDataSource1" DataTextField="Семейство"
```

```

    DataValueField="Код_семейства" Height="24px" Width="158px"
    AutoPostBack="True">
</asp:DropDownList>

```

Здесь ID="Семейства" присвоение элементу имени, runat="server" указание на то, что все операции с элементом управления происходят на сервере, DataSourceID="SqlDataSource1" имя контрола с описанием источника данных элемента управления, который будет рассмотрен ниже, DataTextField="Семейство" и DataValueField="Код_семейства" имена полей данных из таблицы базы данных значения которых соответствует карточке коллекции ("Код_семейства") и отражается в элементе управления ("Семейство").

Контроль SqlDataSource с именем SqlDataSource1 описывает источник данных для рассматриваемого элемента управления DropDownList и имеет вид.

```

<asp:SqlDataSource ID="SqlDataSource1" runat="server"
    ConnectionString="<%$
ConnectionString:Гербарий_ЛТАConnectionString %>"

```

```

    SelectCommand="SELECT 0 AS Код_семейства, ' Выбор семейства' AS
Семейство UNION SELECT DISTINCT Виды.[Код семейства] AS
Код_семейства, Семейства.Семейство FROM Коллекция INNER JOIN
Виды ON Коллекция.[Код вида] = Виды.[Код вида] INNER JOIN Рода ON
Виды.[Код рода] = Рода.[Код рода] INNER JOIN Семейства ON Рода.[Код
семейства] = Семейства.[Код семейства] WHERE (@CodStrany = 0 OR
Коллекция.[Код страны] = @CodStrany) AND (@CodNasP = 0 OR
Коллекция.[Код пункта] = @CodNasP) AND (@CoNameKol = 0 OR
Коллекция.[Код наименования коллекции] = @CoNameKol) AND
(@CodKollect = 0 OR Коллекция.[Код коллектора] = @CodKollect) AND
(@DtSbora = ' ' OR Коллекция.[Дата сбора] LIKE '%' + @DtSbora + '%')
ORDER BY 2">

```

```

<SelectParameters>
    <asp:ControlParameter ControlID="Страна" DefaultValue="0"
Name="CodStrany"
    PropertyName="SelectedValue" />
    <asp:ControlParameter ControlID="НасПункт" DefaultValue="0"
Name="CodNasP"
    PropertyName="SelectedValue" />
    <asp:ControlParameter ControlID="Наименования" DefaultValue="0"
Name="CoNameKol" PropertyName="SelectedValue" />

```

```

    <asp:ControlParameter ControlID="Коллекторы" DefaultValue="0"
Name="CodKollect"
    PropertyName="SelectedValue" />
    <asp:ControlParameter ControlID="ГодСбора" DefaultValue=" "
Name="DtSbora"
    PropertyName="Text" />
</SelectParameters>
</asp:SqlDataSource>
    Здесь ConnectionString="<%$
ConnectionString:Гербарий_ЛТАConnectionString %>" имя строки с
описанием подключения к базе данных, которое имеет следующий вид:

```

```

<connectionStrings>
    <add name="Гербарий_ЛТАConnectionString" connectionString="Data
Source=GORVICAL;Initial Catalog=Гербарий_ЛТА;Integrated
Security=True"
    providerName="System.Data.SqlClient" />
</connectionStrings>

```

В этом тэге определяется имя сервера (GORVICAL) и базы данных (Гербарий_ЛТА). Для подключения к другому серверу достаточно сменить имя сервера и WEB форма будет работать с базой на другом сервере. В свойстве SelectCommand тега SqlDataSource задаётся SQL запрос на выборку семейств, которые присутствуют в коллекции. Запрос имеет параметры, позволяющие настраивать результаты в зависимости от страны, населенного пункта, имени коллекции, коллектора и времени сбора. Это означает, что в момент выбора семейства в списке выбора появляются только те из них, которые соответствуют ранее выбранным параметрам. Например, если сначала выбрать коллектора, то в списке семейств появятся семейства собранные этим коллектором. Этот же подход относится и ко всем остальным условиям поиска. В результате в списках каждого из условий поиска появляются только те варианты, которые соответствуют ранее выбранным условиям и присутствуют в коллекции. Для реализации подобной зависимости для страны, населенного пункта, имени коллекции, коллектора требуется несколько другой подход. Это обусловлено тем, что в базе для каждой карточки хранится только код вида, по которому схема базы позволяет легко найти род и семейство. Однако, если в качестве части условия задано только семейство, то по нему в результат выборки нужно включить все виды этого семейства, то есть в условие запроса нужно включить подчиненный запрос. Для этого предлагается вместо запроса в свойстве

SelectCommand использовать хранимые процедуры. Хранимые процедуры имеют широкие возможности и могут включать в себя запросы и операторы. Кроме того, они выполняются на стороне сервера и возвращают результат с учетом заданных параметров. Так для контроля осуществляющего выбор наименования коллекции и имеющего в сайте следующий тег

```
<asp:DropDownList ID="Наименования" runat="server"
  DataSourceID="SqlDataSource7" DataTextField="Наименование
коллекции"
  DataValueField="Код наименования коллекции" Width="230px"
  AutoPostBack="True">
</asp:DropDownList>
```

используется источник данных SqlDataSource7 имеющий следующий вид

```
<asp:SqlDataSource ID="SqlDataSource7" runat="server"
  ConnectionString="<%=
ConnectionStrings:Гербарий_ЛТАConnectionString %>"
  SelectCommand="NameKollect"
  SelectCommandType="StoredProcedure">
  <SelectParameters>
    <asp:ControlParameter ControlID="Семейства" DefaultValue="0"
Name="CodSem"
      PropertyName="SelectedValue" Type="Int32" />
    <asp:ControlParameter ControlID="Рода" DefaultValue="0"
Name="CodRod"
      PropertyName="SelectedValue" Type="Int32" />
    <asp:ControlParameter ControlID="Виды" DefaultValue="0"
Name="CodVid"
      PropertyName="SelectedValue" Type="Int32" />
    <asp:ControlParameter ControlID="Страна" DefaultValue="0"
Name="CodStrany"
      PropertyName="SelectedValue" Type="Int32" />
    <asp:ControlParameter ControlID="Коллекторы"
DefaultValue="0" Name="CodKollect"
      PropertyName="SelectedValue" Type="Int32" />
  </SelectParameters>
</asp:SqlDataSource>
```

Как видно из этого декларативного описания, определен тип команды SelectCommandType="StoredProcedure", который указывает на то,

что источником данных является хранимая процедура *NameKollect* (*SelectCommand="NameKollect"*). Процедура имеет параметры: «Семейства», «Рода», «Виды», «Страна», «Коллекторы». Это значит, что если в условии поиска определен хотя бы один из этих параметров, наименование коллекции будет выбираться из списка, который сформирован из карточек коллекции, включающих значения этих параметров. Такой подход гарантирует, что в списке выбора будут присутствовать значения наименований коллекции имеющиеся в базе данных. Хранимая процедура, которая осуществляет подобный выбор имеет следующий вид (пояснения к процедуре приведены в комментариях):

```
ALTER PROCEDURE dbo.NameKollect
    /* входные параметры процедуры */
    (
        @CodSem int,
        @CodRod int,
        @CodVid int,
        @CodStrany int,
        @CodKollect int
    )
AS
/* переменные для внутреннего использования*/
/* @FragUsl – текст сформированного условия для запроса */
/* @strSQL – текст запроса */
/* @ParmDef - текст перечня параметров для процедуры sp_executesql
выполняющей запрос */
/* @Perecl – переключатель определяющий какое условие формировать */
    DECLARE @FragUsl varchar(120), @strSQL
nvarchar(2000),@ParmDef nvarchar(100),@Perecl int
/* первоначально строка с условием должна быть пустой */
    SET @FragUsl = "
/* в строке параметров для выполнения запроса указываются код
коллектора, код страны и код систематики */
    SET @ParmDef = '@CodKol int,@CodStr int,@CodSist int'
/* если входной параметр код вида больше нуля */
    IF @CodVid > 0
/* то в фрагмент условия добавить выражение отбора по коду вида */
/* и установить переключатель в 1 */
    BEGIN
        SET @FragUsl= ' And Коллекция.[Код вида] = @CodSist'
        SET @Perecl=1
```

```

        END
    ELSE
/* иначе, если код род больше 0 */
        BEGIN
            IF @CodRod > 0
/* то во фрагмент условия добавить отбор всех кодов вида относящихся к
роду, переданному во входном параметре и установить переключатель в 2
*/
                BEGIN
                    SET @FragUsl= ' And (Коллекция.[Код вида] In (SELECT
Виды.[Код вида] FROM Виды WHERE Виды.[Код рода] = @CodSist))'
                    SET @Perecl= 2
                END
            ELSE
/* если не заданы вид и род, а код семейства больше 0 */
                BEGIN
                    IF @CodSem > 0
/* установить в фрагмент условия добавить выражение отбора всех кодов
вида относящихся к семейству, переданному во входном параметре и
установить переключатель в 3 */
                        BEGIN
                            SET @FragUsl= ' And (Коллекция.[Код вида] In (SELECT
Виды.[Код вида] FROM Виды WHERE Виды.[Код семейства] =
@CodSist))'
                            SET @Perecl= 3
                        END
                    END
                END
            END
/* если связанные с систематикой параметры не заданы */
            IF @FragUsl= ''
                BEGIN
/* то выполнить запрос по условию отбора – коллектор, страна */
                    SELECT DISTINCT Коллекция.[Код наименования коллекции],
[Именованные коллекции].[Наименование коллекции]
FROM      Коллекция INNER JOIN
          Виды ON Коллекция.[Код вида] = Виды.[Код вида] INNER
JOIN
          [Именованные коллекции] ON Коллекция.[Код наименования
коллекции] = [Именованные коллекции].[Код наименования коллекции]

```

```

WHERE (Коллекция.[Код наименования коллекции] = 0) OR
((@CodKollect = 0 OR Коллекция.[Код коллектора] = @CodKollect) AND
(@CodStrany = 0 OR Коллекция.[Код страны] = @CodStrany))
    END
    ELSE
/* если заданы параметры с систематикой, то сформировать текст запроса с
учетом ранее сформированного фрагмента условия */
    BEGIN
        SET @strSQL = 'SELECT DISTINCT Коллекция.[Код
наименования коллекции], [Именованные коллекции].[Наименование
коллекции]
FROM      Коллекция INNER JOIN Виды ON Коллекция.[Код вида] =
Виды.[Код вида] INNER JOIN
[Именованные коллекции] ON Коллекция.[Код наименования коллекции]
= [Именованные коллекции].[Код наименования коллекции]
WHERE (Коллекция.[Код наименования коллекции] = 0) OR ((@CodKol
= 0 OR Коллекция.[Код коллектора] = @CodKol) AND (@CodStr = 0 OR
Коллекция.[Код страны] = @CodStr)' + @FragUsl + ')'
/* сформировать обращение к процедуре, которая выполняет текст
сформированного запроса в зависимости от значения переключателя */
        IF @Perecl= 1
            exec sp_executesql @strSQL , @ParmDef,
@CodKol=@CodKollect, @CodStr=@CodStrany, @CodSist=@CodVid
        ELSE
            IF @Perecl= 2
                exec sp_executesql @strSQL , @ParmDef,
@CodKol=@CodKollect, @CodStr=@CodStrany, @CodSist=@CodRod
            ELSE
                exec sp_executesql @strSQL , @ParmDef,
@CodKol=@CodKollect, @CodStr=@CodStrany, @CodSist=@CodSem
        END
    RETURN

```

Список наименований коллекций полученный с помощью этой хранимой процедуры появляется при нажатии кнопки выбора соответствующего контроля.

Подобный подход позволяет исключить случаи поиска в коллекции экземпляров карточек с отсутствующими параметрами.

Использование хранимых процедур подобного рода задачах (создание условий с взаимозависимыми параметрами) является единственным на мой взгляд решением, которое точно отвечает поставленной задаче.

ТЕСТИРОВАНИЕ ЗНАНИЙ СТУДЕНТОВ В СРЕДЕ MYTEST X

MyTest X это система программ для создания тестов, проведения компьютерного тестирования, сбора и анализа результатов тестирования, выставления оценки по указанной в тесте шкале.

Она считается лучшей бесплатной российской системой программ такого рода

Разработчик системы: Башлаков Александр Сергеевич, учитель информатики, г. Унеча, Брянской обл.

Наиболее полная и актуальная справочная информация расположена на сайте программы <http://mytest.klyaksa.net>. Имеются обучающие видеоролики по установке и работе с программой. Так же на сайте организуется банк тестов. Вы можете опубликовать в нем свои тесты.

Для скачивания предлагаются следующие дистрибутивы:

- **MyTestXSetup.exe** – Файл установки программы. (рекомендуется!). Инсталлятор поможет вам легко установить программу, а также создать значки и ярлыки, ассоциировать файлы, указать нужный язык интерфейса.

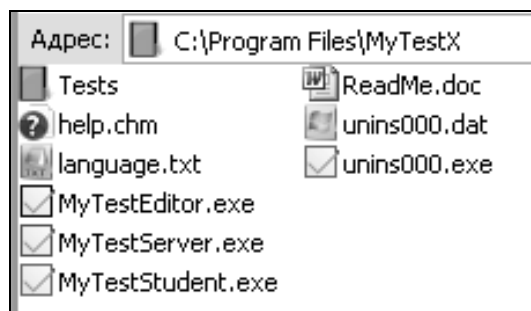
- **MyTestX.rar, MyTestX.zip** – Архивы с программой (для установки вручную). Достаточно распаковать архив в нужную папку на вашем компьютере. Создать ярлыки и ассоциировать файлы можно позже через меню настройки программы или самостоятельно. Информация о настройке программы есть в разделе справка.

- **MyTestStudentTerminal.rar** – Модуль тестирования для терминальных классов. Замените файл MyTestStudent.exe в папке с программой на файл из этого архива.

- **ConverterMyTestX0.exe** – Утилита для преобразования тестов из формата MyTest 3.0.4 в новый формат MyTest X

- **import_dies.rar** – Программа для импорта тестов из текстовых файлов специального формата (пример в файле) в формат программы MyTest X 10.0.

Для работы в локальном режиме проще всего установить систему с помощью файла **MyTestXSetup.exe** как обычную прикладную программу. В результат получим:



где

MyTestEditor.exe – модуль редактора тестов.

MyTestStudent.exe – модуль тестирования.

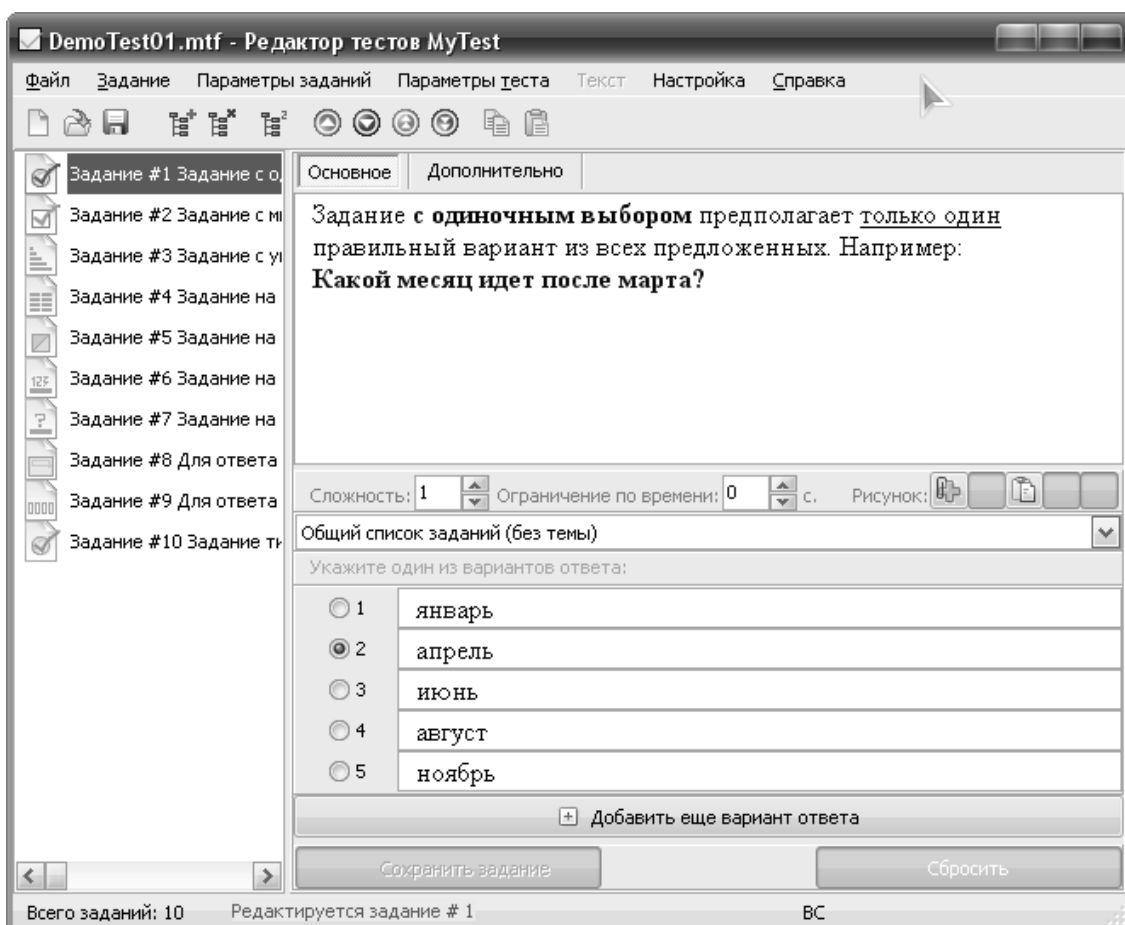
MyTestServer.exe – модуль журнала тестирования.

Help.chm – файл справки.

Tests – папка с демонстрационными тестами.

Редактор тестов (MyTestEditor.exe)

С помощью редактора можно создать либо новый тест, либо редактировать существующий. Интерфейс модуля редактора тестов показан на рисунке ниже.



Задания (вопросы), параметры тестирования, звуки и изображения к заданиям для каждого отдельного теста – все хранится в одном файле теста. Никаких баз данных: один тест – один файл. Файл с тестом зашифрован и сжат.

Поддерживает *девять* типов заданий (см. Help.chm). К каждому заданию могут быть прикреплены рисунки и звуковые файлы.

Ввод текста заданий (вопросов) и ответов к ним производится с помощью встроенного текстового редактора (похожего на WordPad), что позволяет форматировать задания и ответы.

Предусмотрены несколько независимых друг от друга режимов тестирования: **Обучающий** – при ошибке тестируемый получает соответствующее сообщение и может посмотреть объяснение задания (если оно вами задано). Ошибочные и верные варианты подсвечиваются разными цветами. Можно задать объяснение ошибки, которое будет показано пользователю. **Штрафной** – при ошибке у тестируемого будут отниматься баллы. В этом режиме дополнительно появляется кнопка "Я не знаю...". Щелчок по ней присваивает за задание нуль баллов. Этот режим работает по-разному в зависимости от настроек оценивания (засчитывать 100% верные ответы или нет).. **Свободный** - позволяет перемещаться по заданиям в любом порядке. В любой момент можно вернуться к заданию, если на него еще не дан ответ. Дополнительно появляется кнопка "Пропустить" – переход к следующему заданию без ответа (позже можно вернуться) и список заданий (в котором по-разному отмечены просмотренные и выполненные задания), позволяющий перейти к любому невыполненному заданию. **Монопольный** - окно программы занимает весь экран и его невозможно свернуть.

Каждый тест может содержать:

- **Заголовок** теста. **Автор** (составитель) теста. **Описание** теста.
- **Инструкцию тестируемому** - сообщение, которое будет показано тестируемому в начале теста. Инструкция может содержать форматированный текст, рисунки и формулы.
- **Оценивание.** Оценка тестируемого вычисляется по баллам. Система оценки может быть любой – от двухбалльной (зачет/незачет) до 100-балльной. Для оценки можно указать название - тогда выводится не цифра, а название (полезно для зачет/незачет). В редакторе для удобства имеются часто применимые шаблоны оценивания, вы можете использовать их, а можете задать свои.
- **Порядок вопросов.** Обычный или случайный. Задает, в каком порядке задания будет предложены тестируемому. Для случайного порядка перемешивание заданий происходит при каждом запуске теста.

- **Порядок ответов.** Обычный или случайный. Задает, в каком порядке тестируемому будет предложены варианты ответов. Если включен случайный порядок, то у разных тестируемых будет различный порядок вариантов ответа.

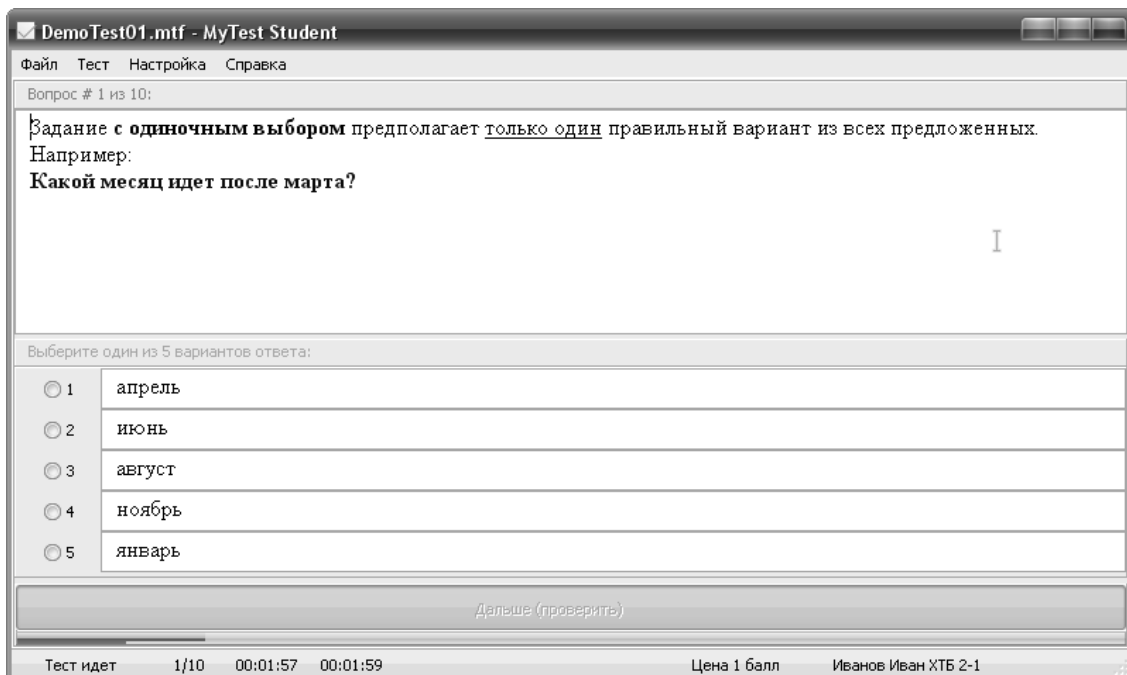
- **Темы (группы) заданий.** Темы заданий предназначены для ограничения количества заданий для тестируемого. Можно указать, сколько именно заданий из какой темы будет задано. При ограничении количества заданий они выбираются случайным образом.

- **Ограничение по времени.** Вы можете ограничить не только время обдумывания каждого задания, но и время всего теста. Ограничение для задания указывается при редактировании заданий, ограничение для всего теста в меню **Параметры тестирования**.

- **Пароли.** Вы можете задать пароль на открытие теста для тестирования, открытие теста для редактирования, на запуск тестирования, для сохранения защищенных результатов.

Модуль тестирования (MyTestStudent.exe)

Модуль тестирования – это модуль программы MyTest в котором учащиеся проходят тестирование. Программа проста в использовании и имеет удобный интерфейс. Но, не смотря, на всю ее простоту позволяет эффективно организовать тестирование, сохранение и отправку результатов преподавателю. Интерфейс модуля тестирования с открытым тестом **DemoTest01.mtf** показан на рисунке ниже.



Модуль тестирования может находиться в одном из трех состояний:

тест не выбран, тест выбран, тест идет. Текущие состояние отображается в строке состояния.

Выбор правильного варианта ответа производится либо щелчком мыши, либо клавишами на клавиатуре.

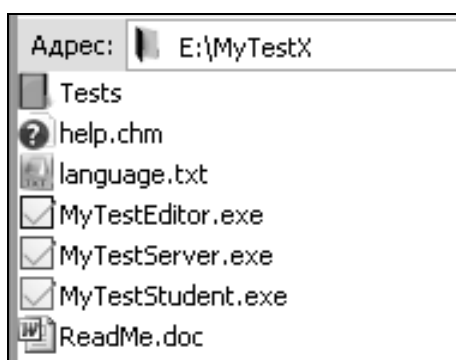
Для перехода к следующему вопросу необходимо нажать кнопку "Дальше" либо клавишу ENTER или ПРОБЕЛ.

Нижняя часть окна программы предоставляет информацию об количестве вопросов в тесте и количестве заданных вопросов (текущий/всего), времени затраченного на весь тест и на текущее задание, режиме тестирования, имени тестируемого. Если включено ограничение по времени, то так же будет отображаться оставшееся время на обдумывания задания или всего теста. После окончания выделенного на обдумывание времени выводится соответствующее сообщение и производится переход к следующему вопросу (или заканчивается тест и выводятся результаты). Если на данный момент был выбран правильный вариант ответа, то он будет принят. Если время обдумывания заканчивается оно будет выделено красным цветом. При "Свободном режиме", если для заданий задано ограничение по времени, то оно запоминается. Например, для задания стоит ограничение 30 секунд. Как бы тестируемый не выбирал задания, суммарное время показа этого задания не может превысить указанное значение.

Текст вопросов и вариантов ответа защищен от копирования. Ни с помощью комбинации клавиш, ни с помощью меню скопировать его нельзя.

Работа в сетевом режиме

Для работы в сетевом режиме необходимо распаковать любой из архивов **MyTestX.rar** или **MyTestX.zip** в любую папку на компьютере, выбранном в качестве сервера, например:



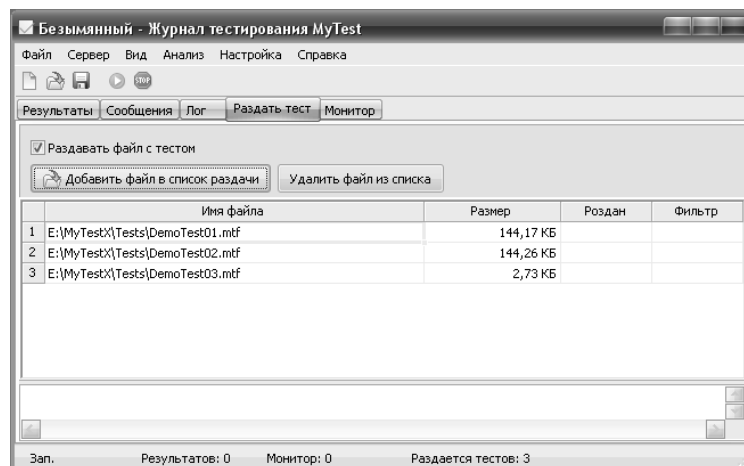
Настройка работы в этом режиме подробно описана в файле справки.

Журнал тестирования (MyTestServer.exe)

Журнал (сервер) – модуль программы MyTest, позволяющий

централизованно принимать и обрабатывать результаты тестирования, раздавать тесты посредством компьютерной сети. Для отправки и получения результатов, отправки файлов с тестами используется протокол Интернета TCP/IP.

Интерфейс модуля Журнала тестирования показан на рисунке ниже. Настройка работы в его среде подробно описана в файле справки.



На вкладке "**Сообщения**" отображается информация о начале или завершении тестирования (например, 21.02.2009 12:11:26 192.168.3.5: Иванов Иван (PC05) начал тест: "Демонстрационный тест №2" (C:\MyTest\ДемоТест_0002.mtf)). Указывается дата и время получения сообщения, имя и IP-адрес компьютера ученика, имя тестирующегося (то которое он ввел при запуске теста).

На вкладке "**Раздать тест**" можно указать какие файлы с тестами будут получать тестируемые по команде **Файл – Получить по сети** (или автоматически при соответствующей настройке). Эта возможность экономит время преподавателя – нет необходимости копировать файл на компьютеры учеников. Еще одно преимущество – файл с тестом не сохраняется на компьютерах тестируемых, у них нет возможности открыть его для подсматривания, поэтому не требуется задавать пароли. Если в список для раздачи тестов добавлен только один файл, то ученик при запросе теста (**Файл - Получить по сети...**) откроет этот файл. Если же в список добавлено несколько файлов, то студент при запросе теста увидит окно, предлагающее выбрать какой именно файл он желает получить. После выбора файла из этого списка он будет загружен.

На вкладке "**Монитор**" выводится информация о процессе тестирования. Вы можете видеть какие компьютеры (и студенты) выполняют тест (и какой тест) и сколько заданий выполнил ученик (+ сколько из них верно или ошибочно, % результативности), сколько баллов

(максимум, возможно было бы набрать за эти ответы, набрано, % предварительный результат).

Ю.А. Жук, старший преподаватель

ПРОБЛЕМЫ РАЗРАБОТКИ И ИСПОЛЬЗОВАНИЯ ДИСПЛЕЙНЫХ ФОРМ НАГЛЯДНОСТИ

За последние десятилетия глобальным образом изменилась информационно-образовательная среда. В условиях реформ в сфере высшего образования, вполне естественным, является стремление восполнить недостаток учебной информации через обращение к электронным ресурсам. Несмотря на очевидные достижения этой отрасли, многие обучающие системы не обладают возможностями адаптации, а также не учитывают психологических особенностей восприятия видеоинформации обучаемыми, что достаточно сильно отражается на эффективности процесса обучения.

Познавательная деятельность человека никогда не протекает, опираясь лишь на одну изолированную модальность (зрение, слух, осязание); любое предметное восприятие является результатом полимодальной деятельности. Комбинированное воздействие визуальной и аудиоинформации дает наилучшие результаты. Отсюда вытекает роль мультимедиа элементов, которые обеспечивают полисенсорность обучения с задействованием практически всех каналов восприятия информации человеком, что обеспечивает качественно новый уровень обучения.

Знание свойств дисплейной мультимедийной наглядности помогают квалифицированно использовать её в обучении. Опираясь на них, можно усилить эффект психолого-педагогического воздействия, реализовать намечаемые дидактические и методические цели.

Визуальная среда на экране монитора является искусственной, по многим параметрам отличающейся от естественной. Естественным для человека является восприятие в отраженном свете, а на экране монитора информация передается с помощью излучающего света. Таким образом, эффективность восприятия информации с дисплея, монитора компьютера зависит от успешности согласования индивидуальных качеств обучаемых с параметрами зрительной или слуховой подачи учебного материала. Ошибки восприятия могут быть связаны с искажениями информации со стороны как человека, так и со стороны неудовлетворительного технического решения продукта, а также несоответствия способа кодирования информативных элементов (языка экранной образности и его

составляющих – изображения предметов, соотношение статических и динамических элементов образа, цифры, надписи, рамки, их цвет, колорит, форма, последовательность и т.д.) возможностям восприятия учащегося и переработки им информации.

В ходе анализа литературы по данной теме, а также обобщения результатов нашей экспериментальной работы была определена система общих требований предъявляемых к дисплейной наглядности:

1. Изображение на дисплее должно быть построено так, чтобы нужная для усвоения информация отображалась фигурой, а фон выделял и подчеркивал те особенности информации, которые в данный момент раскрывают сущность учебного материала. При этом:

- изображение объекта должно занимать площадь всего кадра, если только дидактическими целями и задачами не предусмотрено проведение дальнейшей работы с данным изображением;

- главные элементы должны быть выделены, равнозначные – соответственно оформлены. По возможности, необходимо уменьшить количество в данный момент второстепенной, не рассматриваемой информации;

- при создании контуров объектов, взаимного расположения элементов изображения необходимо обеспечить достаточную толщину линий, при этом главные связи и основные элементы, на которые необходимо обратить внимание, должны быть выделены. Но следует учитывать, что динамичность и яркость деталей затрудняет выделение главного и студент становится невосприимчивым к содержательной информации;

- комфортность восприятия информации с экрана монитора достигается при равномерном распределении яркости в поле зрения;

- необходимо учитывать, насколько проблематично отделение воспринимаемого объекта от окружающего его поля восприятия (отношения фигуры и фона). В связи с этим следует учитывать принципы восприятия, выдвинутые сторонниками гештальтпсихологии (законы прегнантности, или "хорошей формы").

- При подборе цвета, используемого в наглядности, для дисплея с цветным экраном важно знать, как цвет влияет на психику человека и восприятие им информации:

- к взаимодополняющим цветам относятся три пары: красно-зеленый, желто-фиолетовый, сине-оранжевый; при таком сочетании цветов не возникает новых оттенков, а происходит лишь взаимное повышение насыщенности и яркости;

- информационная роль цвета выступает тогда, когда он несет знания о существенных свойствах предметов, при этом для выделения

информации одного и того же типа должны использоваться одни и те же цвета;

- насыщенные цвета должны применяться только для выделения критической, требующей немедленной реакции информации, которую сложно выделить другим способом;
- выбор цвета должен подчиняться основному типу сигнализации, например, красный-опасность, желтый – предупреждение, зеленый – все в порядке и т.д. (ГОСТ);
- рекомендуемое ограничение цветовой гаммы 5 ± 2 ;
- необходимо обеспечить максимальную контрастность, что достигается подбором цветов фона и изображения;

3. Совершенно необходимым, хотя и не достаточным условием того, чтобы информация была воспринята, является приход к органам чувств интенсивных, четких, неискаженных сигналов, соответствующих характеристикам органов чувств, особенностям человеческого восприятия.

Следует учитывать акустику и остроту зрения студентов, находящихся в глубине аудитории, в случае применения небольших мониторов. Острота зрения в большой степени определяется особенностями зрительного анализатора. Однако в некоторых пределах воздействие находится во власти педагога: увеличение яркости фона увеличивает остроту зрения.

Для оптимизации изучения представления информации на экране компьютера рекомендуется использование логических ударений. Логическими ударениями принято называть психолого-аппаратные приемы, направленные на привлечение внимания пользователя к определенному объекту. Психологическое действие логических ударений связано с уменьшением времени зрительного поиска и фиксации оси зрения по центру главного объекта. Наиболее часто используемыми приемами для создания логических ударений являются:

- изображение главного объекта более ярким цветом,
- изменение размера, яркости, расположения,
- выделение проблесковым свечением.

На восприятие информации с дисплея влияет утомление. Сенсорные функции при зрительном утомлении отчетливо снижаются. При длительной работе целесообразно чередовать периоды зрительной нагрузки с периодами, когда работает главным образом слуховое восприятие. Восприятие – активный процесс. Люди могут увидеть разное, даже рассматривая один и тот же объект. Это объясняется тем, что восприятие зависит не только от сигналов, пришедших в мозг от рецепторов, но и от того, чего ждет субъект, осуществляющий вероятностный прогноз. Чем меньше слушатель ждет того или иного

сигнала, тем большая четкость требуется от этого сигнала, чтобы он был принят неискаженным.

Необходимо также учитывать проблему выбора количества кадров для одного занятия, разбиения их на информационные блоки, определение объема текста, использования графики и анимации и др., а именно:

– демонстрация одного кадра должна занимать от одной до пяти минут, т.к. слишком частая смена кадров не позволяет осмысливать их содержание, особенно в контексте с речевым изложением, а использование небольшого количества кадров (и их редкая смена) обедняет информацию. Кроме того, смена кадра является одним из способов привлечения внимания аудитории, не дает занятию становиться монотонным;

– экспериментально было доказано, что время необходимое для выполнения операции сравнения наглядного материала, зависит от формы кодирования информации (при графическом кодировании - это время меньше, чем при числовом);

– одним из важных аспектов дискретного предъявления информации является размер и насыщенность информационных блоков. Так отмечается, что "при числе объектов от 2 до 9 их можно сосчитать, при 9-40 объектах воспринимаются мелькания, при более чем 40 объектах они неразличимы";

– восприятие зависит не только от количества элементов, но и от целого ряда параметров, таких как уровень подготовки студентов, сложность преподаваемого материала, продолжительность объяснения и др.

4. Важнейшее требование, предъявляемое к тексту (точнее называть его медиатекстом), который используется в дисплейных формах наглядности – это соответствие дидактическому принципу научности. Скорость и точность восприятия с дисплея текстовой информации зависят от объема текста, характеристики букв, интервалов между ними, межстрочных интервалов, длины и количества строк. Поэтому при оформлении медиатекста необходимо соблюдать следующие условия:

- большое количество текста на изображении уменьшает эффективность восприятия наглядного пособия, поэтому оно должно содержать тот минимум текста, который необходим для обозначения отдельных объектов, их элементов;

- при написании текста необходимо максимально подчеркивать различия подобных по направлению букв (С-О, Б-В и т.д.) и особенно цифр (т.к. на восприятие цифр в большинстве случаев автокоррекция не распространяется);

- оптимальное соотношение высоты букв и их ширины 2:3, расстояние между буквами 0,2-0,3 их ширины, а расстояние между строками 1-2 высоты букв (ГОСТ на видеоинформацию);
- плохо воспринимаются вертикальные надписи и расположенные по окружности;
- наиболее удобочитаем черный шрифт на белом фоне, затем черный набор на всех цветных планшетах, неудобочитаем желтый, белый на дискретном фоне;
- выделение шрифта другим цветом при чтении текста способствует закреплению материала в долговременной памяти.
- Необходимо учитывать ряд иных требований выдвигаемых к дисплейной наглядности:
 - *эргономические* – требования к качеству отображения информации, обеспечение оптимального управления, включая развитую систему навигации и достаточную скорость извлечения информации, обеспечения устойчивости к ошибочным действиям пользователя;
 - *санитарно-гигиенические* – допустимое время работы с мониторами, размещение компьютеров в учебном помещении для снижения утомляемости и сохранения здоровья обучающихся;
 - *требования эстетизации* – соответствие современному дизайну и эстетическим нормам.

Для решения задач улучшения визуализации и формирования диалогового взаимодействия используются современные средства мультимедиа. Развитие мультимедийных систем, т.е. многопланового представления информации, включая графику высокого разрешения с реалистичной цветовой палитрой, анимированным изображением, живое, в том числе полноэкранное, видео, позволило поднять презентабельность компьютерных обучающих программ, на более высокий качественный уровень. При проектировании и создании обучающих программ требуется учет психологических принципов взаимодействия человека и компьютера.

Использование мультимедиа в образовательных технологиях является полезным и плодотворным, благодаря присущим ей качествам интерактивности, гибкости, адаптивности. А интеграция различных форм учебной информации в дисплейной наглядности, предоставляет возможность учитывать индивидуальные особенности познавательной деятельности обучающихся, способствует активации всех анализаторов, в результате чего возникает богатая эмпирическая основа для обобщающе-аналитической мыслительной деятельности студента.

РАСПОЗНАВАНИЕ ГРАНИЦ ДЕФЕКТОВ НА ПОВЕРХНОСТИ ПИЛОМАТЕРИАЛА

Выделение границ объекта на полученном изображении, как это было описано в [1], является одной из главных задач, определяемых основными критериями распознавания дефектов древесины. Более того, из приведенных критериев распознавание по геометрическим формам дефектов идет первой по счету задачей. От того, насколько правильно определяются границы дефектов, зависит правильность дальнейшего распознавания не только по форме, но и по диапазону яркостей и количеству пикселей, входящих в эти границы.

Алгоритм операций предшествующих распознаванию по формам выбран следующий:

1. Перевод изображения (полноцветного, палитрового или в оттенках серого) в бинарное.
2. Выделение границ пиломатериала.
3. Удаление из изображения неинформативных частей, не относящихся к пиломатериалу.
4. При необходимости выравнивание освещения.
5. Удаление тех объектов, чья длина/ ширина/ диаметр/ площадь меньше пороговых или подсчет их количества на определенной площади с последующим удалением в случае неравномерности их распределения.
6. Выделение границ объектов (предполагаемых дефектов) на изображении пиломатериала.
7. При необходимости сглаживание границ объектов путем выбора большей толщины границ выделения.
8. Вписывание полученных границ в рамки определенных размеров
9. Распознавание геометрических фигур, создаваемых получаемыми границами.

Разберем пример нахождения границ дефектов на участке пиломатериала с помощью пакета Image Processing Toolbox среды MatLab7.1. Данный пакет содержит много полезных функций по обработке изображений, в том числе по морфологическим операциям над бинарным изображением [2], т.к. большинство операций будет проходить именно над таким представлением изображения.

Опустим сразу второй и третий пункты вышеописанного алгоритма, потому что определение границ пиломатериала сходно определению

границ дефектов, и возьмем фотографию участка пиломатериала без участков ленты транспортера (рис. 1).



Рис. 1. Фотография участка пиломатериала с двумя дефектами.

Сначала переведем полноцветное изображение в бинарное (рис. 2а). Отсечение по порогу яркости выбирается в зависимости от освещенности, породы древесины и ее свойств (например, влажности). На полученном бинарном изображении имеется некоторый шум из-за теней от неровностей поверхности пиломатериала и более ярко окрашенных древесных волокон. С помощью функции `bwareopen` удалим те связанные объекты, чья площадь, в данном случае, меньше 2048 пикселей (рис. 2б). Порог по количеству пикселей выбирается в зависимости от размера получаемого изображения и минимальных размеров выбранных дефектов.

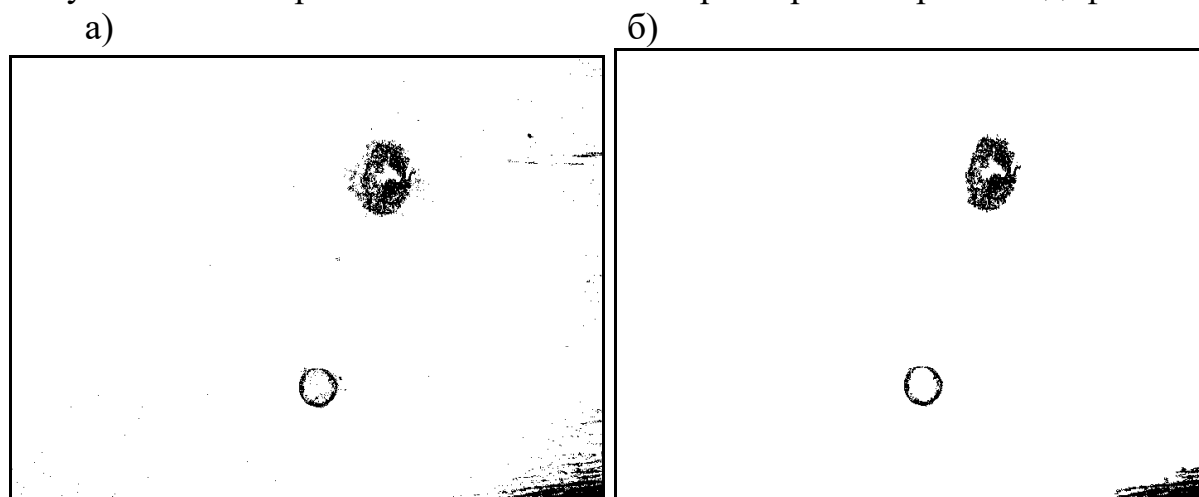


Рис. 2. а - зашумленное бинарное изображение и б - оно же без шумов

На рис. 2а и 2б в правом нижнем углу в результате неравномерности освещения присутствует участок края пиломатериала, не относящийся к дефектам. В цифровой обработке можно, но не рекомендуется использовать функцию выравнивания освещения. На практике эта проблема решается с помощью правильной настройки системы освещения. В данном случае воспользуемся функцией `imclearborder`, осуществляющую подавление световой структуры по краям изображения. Разумеется, ее надо осторожно использовать в реальной системе, иначе будут удалены те дефекты, протяженность которых может быть больше размеров фотографии, например, трещин, или те, которые частично попали в объектив камеры. Эту функцию можно применять в качестве альтернативы функциям выделения и удаления границ частей изображения, не относящихся к пиломатериалу. Таким образом, инвертировав изображение, из него удаляется край пиломатериала.

Как видно на рис. 3а изображения дефектов имеют довольно рваные размытые границы, а нижний объект имеет еще и незамкнутые границы.

Для устранения этой проблемы воспользуемся функцией поиска связанных областей пикселей объектов `bwlabel` и функцией наращивания бинарного изображения `imdilate` (Также для этой цели можно использовать функцию `imclose`).

В результате получим сглаженные и замкнутые границы изображений дефектов (рис. 3б) и матрицу номеров объектов, по которой можно в последующем присвоить номера объектов для вписанных в рамки определенных размеров границ изображений дефектов.

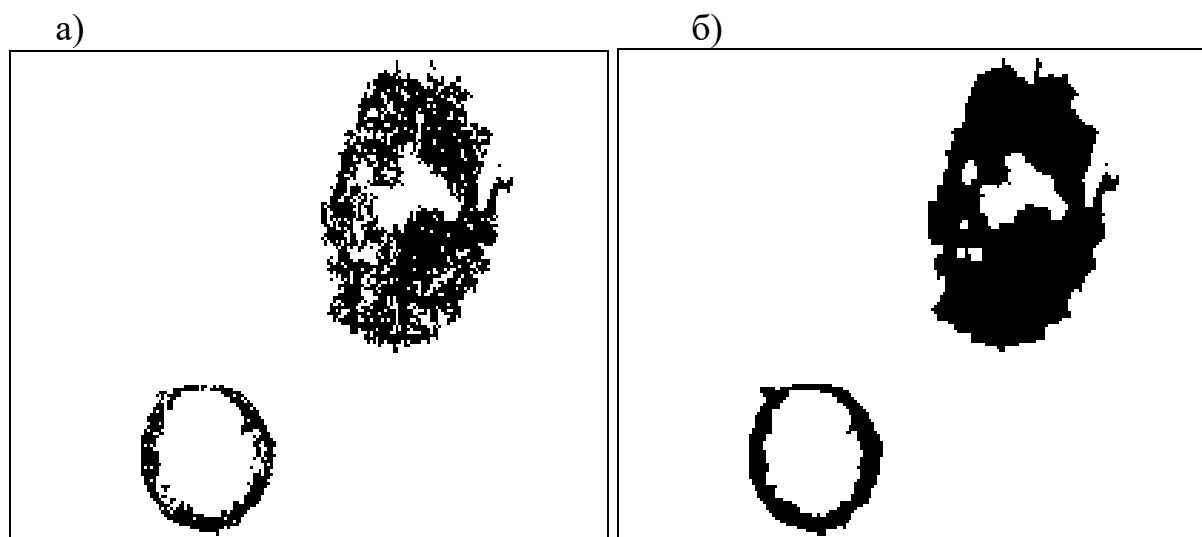


Рис. 3. а - увеличенное изображение дефектов и б - наращение границ дефектов

Также можно провести операцию заливки отверстий в объектах с помощью функции `imfill` (рис. 4а). По полученной матрице в дальнейшем можно будет вырезать из палитрового или полноцветного изображения сами дефекты путем простого перемножения этой матрицы на RGB-компоненты для распознавания по диапазонам яркостей.

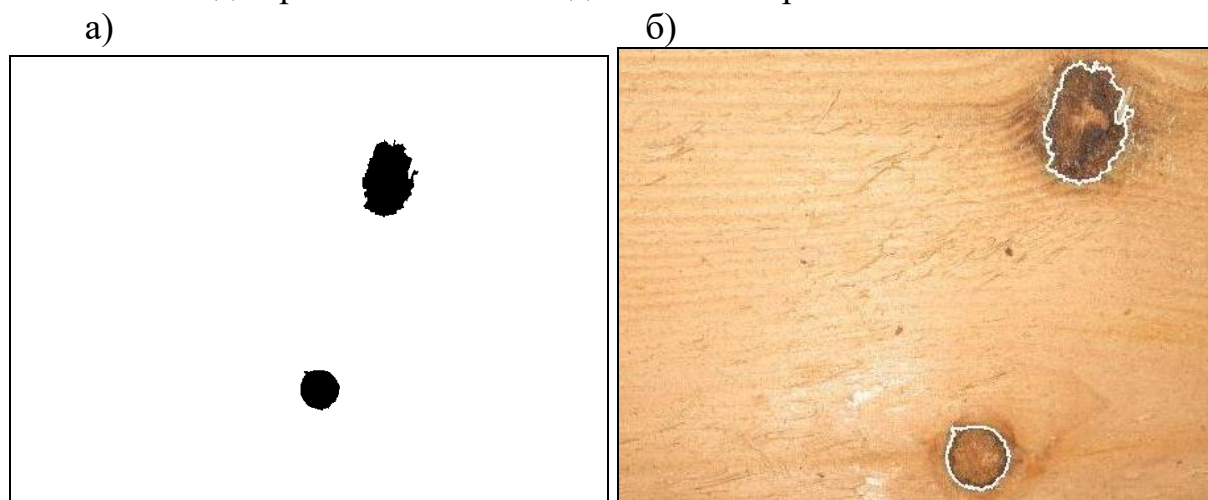


Рис. 4. а - заливка отверстий и б - вычисленные границы дефектов на поверхности п/м

Таким образом, по изображению на рис. 4а, используя функцию отслеживания наружных границ объектов `bwboundaries`, определяем границы объектов и выводим их на цветное изображение (рис. 4б).

Подводя итог, можно отметить, что среда MatLab обладает широчайшим набором функций для создания автоматизированной системы обработки изображений и определения границ объектов.

При использовании пакета Image Acquisition Toolbox можно получить сигналы с фотодатчиков, при использовании Image Processing Toolbox – обработать изображение, а с помощью пакета Neural Network Toolbox – распознать дефекты по полученным формам, гистограммам и текстурным признакам.

Приведенный алгоритм частично или полностью используется в системах промышленного зрения, распознавании изображений и робототехнике. При наличии соответствующей базы дефектов указанный алгоритм можно использовать в любой отрасли промышленности для систем сортировки продукции.

Библиографический список

1. Илющенко В.В. Современные методы дефектоскопии пиломатериалов. Леспроминформ № 1 (67) 2010. СПб, 2010
2. <http://matlab.exponenta.ru/imageprocess/book3/index.php>

ПОИСК УЧЕБНЫХ МАТЕРИАЛОВ ПО МЕТАДАННЫМ

Развитие электронного обучения в последние годы привело к появлению во всемирной сети большого количества электронных учебных материалов, в результате чего стала актуальной проблема эффективного поиска этих материалов.

В данной работе рассматриваются существующие пути решения этой проблемы и предлагается новый подход её решения. В ходе работы:

- изучены существующие стандарты и спецификации, относящиеся к поиску учебных материалов;
- рассмотрены и классифицированы хранилища (репозитории), предназначенные для хранения учебных материалов;
- рассмотрены распространённые системы поиска учебных материалов по метаданным;
- сформулированы критерии, позволяющие сравнить рассмотренные системы;
- произведён сравнительный анализ этих систем, выявивший их основные достоинства и недостатки;
- предложен новый подход к реализации системы, наиболее удовлетворяющей современным потребностям в поиске учебных материалов по метаданным.

Основными стандартами и спецификациями, позволяющими унифицировать поиск учебных материалов по метаданным, являются:

- стандарт IEEE LOM (Learning Object Metadata) – используется для унифицированного описания метаданных, связанных с электронным обучением. Этот стандарт был основан на разработанной ранее спецификации IMS Learning Resource Meta-data, и имеет незначительные отличия от неё. Стандарт описывает древовидную структуру метаданных учебного объекта, состоящую из почти 80 элементов данных. На практике в эту структуру при необходимости могут быть добавлены дополнительные элементы данных, регламентированные разработчиком в качестве так называемых «прикладных профилей». В некоторых странах, где активно применяется электронное обучение, были разработаны такие профили, учитывающие специфику национальных систем обучения (например: UK LOM Core в Великобритании, CanCore в Канаде, ANZ-LOM в Австралии и Новой Зеландии, NORLOM в Норвегии, ISRACore в Израиле, RUS_LOM в России);

- спецификация IMS DRI (Digital Repositories Interoperability) – описывает унифицированную модель взаимодействия с серверами, в базах которых хранятся учебные материалы. В этой спецификации описаны 8 базовых функций, среди которых:

- SEARCH – запрос от клиента к серверу на получение результатов поиска по метаданным базы сервера; GATHER – сбор промежуточным репозиторием метаданных о ресурсах другого репозитория;

- EXPOSE – предоставление репозиторием метаданных в ответ на запросы SEARCH или GATHER;

- REQUEST – запрос на получение ресурса, найдено ранее по метаданным при помощи SEARCH;

- DELIVER – предоставление репозиторием ресурса в ответ на запрос REQUEST;

- ALERT – оповещение репозиторием подписчиков о появлении в его базе новых ресурсов или новых метаданных;

- SUBMIT – отправка ресурса в репозиторий;

- STORE – представление сохраняемого в репозитории ресурса с обеспечением возможности унифицированного доступа к нему.

Для анализа различных существующих хранилищ учебных метаданных в данной работе была использована классификация, разделяющая их на три типа:

- хранилища в рамках систем управления обучением (СУО, например: Moodle, Sakai, ATutor и др.);

- специализированные хранилища с централизованным сервером (LORNET, LRE European Schoolnet, MERLOT, Kursnavet и др.);

- распределённые хранилища учебных материалов.

Для сравнения рассматриваемых систем поиска учебных материалов были использованы следующие критерии:

- поддержка стандартов и спецификаций описания учебных материалов (IMS LRM, IEEE LOM, прикладные профили LOM);

- возможность использования в качестве подсистемы в системе управления обучением (чтобы предоставлять возможность оперативного поиска учебного материала разработчику курса либо учащемуся);

- возможность поиска учебных материалов по метаданным во внутреннем хранилище системы управления обучением;

- возможность поиска учебных материалов в глобальных централизованных хранилищах;

- возможность поиска учебных материалов в распределённых хранилищах;

- гибкость поисковых запросов;
- наличие русскоязычного интерфейса.

Из рассмотренных систем поиска среди организаций, предоставивших для поиска свои сервера учебных материалов, Ariadne, LORNET, MERLOT, Nime, Plone и многих других для сравнительного обзора были отобраны типичные представители разных видов:

- Ariadne GLOBE – глобальная система поиска с централизованным сервером, который собирает, хранит и предоставляет клиентам информацию об учебных материалах, хранящихся на множестве других централизованных серверов разных организаций, принявших участие в этом проекте.

- Merlot Content Repository Module – типичный пример подсистемы поиска для системы управления обучением СУО. Данная подсистема является модулем расширения для LMS Moodle и предоставляет возможность поиска учебных материалов по централизованному хранилищу MERLOT (Multimedia Educational Resource for Learning and Online Teaching).

- LeedsMet Repository Open Search – система поиска учебных материалов по базе университета Лидса (Leeds Metropolitan University).

- Система поиска по библиотеке HyperMethod eLearning CMS – типичная подсистема в рамках системы управления обучением, предоставляющая возможность поиска учебных материалов в её внутреннем хранилище.

- Подсистема поиска для DSpace Repository – экспериментальная разработка сотрудников университета из города Патры, позволяющая искать материалы по метаданным LOM в открытом кроссплатформенном репозитории DSpace.

- u-Library – совместная экспериментальная разработка Российского государственного университета инновационных технологий и предпринимательства и института открытых систем при Ивановском государственном энергетическом университете, представляет собой распределённую систему хранения и поиска учебных материалов. Также эта система может подключаться к централизованным хранилищам через интерфейс UDDI.

Результаты сравнения перечисленных систем по указанным ранее критериям отражены в сравнительной таблице (см. табл. 1).

Т а б л и ц а 1

	GLOBE	Merlot	LeadsMet	DSpace	HM CMS	u-Library
спецификации	LOM	LOM	LOM	LOM	-	RUS_LO M
подсистема СУО	Moodle	ATutor	-	-	HM CMS	-
поиск в хранилище СУО	-	-	-	-	HM CMS	-
поиск в централи- зованном глобаль-ном хранилище	GLOBE	Merlot	LeadsMet	-	-	+
поиск в распределённом хранилище	-	-	-	-	-	+
гибкость запросов	-	-	+	+	+	+
рус.яз. интерфейс	-	-	-	-	+	+

Как видно по таблице, сейчас очень остро стоит проблема с поиском материалов во внутренних хранилищах систем управления обучением, в связи с этим возникает необходимость разработки таких подсистем, по крайней мере для наиболее распространённых открытых систем управления обучением. Кроме того, в основном нынешние системы поиска ориентируются на централизованные хранилища. Поэтому при реализации подсистемы поиска для СУО предлагается отдельно разрабатывать модуль агрегации метаданных, соответствующий спецификации IMS DRI, и отдельно подсистему поиска, обращающуюся к учебному содержанию не напрямую, а через этот модуль. Это позволит не только обеспечить СУО дополнительными данными о ресурсах, хранящихся во внешних репозиториях с поддержкой DRI, но и автоматически создаст распределённую систему доступа к учебным ресурсам за счёт распространения модуля агрегации среди уже эксплуатирующихся СУО.

На рис. 1 показана принципиальная схема данной разработки: в СУО внедряется модуль, собирающий метаданные об учебных материалах как из внутренней базы данных СУО, так и из внешних репозиториях, а встроенная в СУО подсистема поиска осуществляет доступ как к внутренним, так и к внешним материалам через этот модуль.

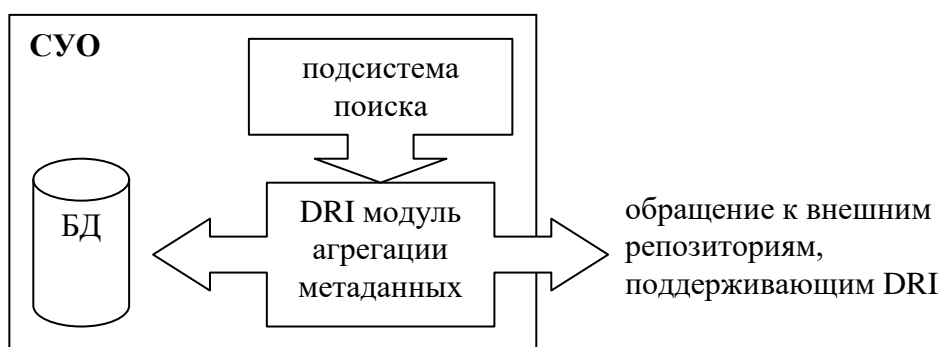


Рис. 1. Схема модулей поиска для СУО

На рис. 2 показана распределённая система хранения и поиска учебных материалов, которая образуется, если внедрить описанный модуль в уже эксплуатирующиеся системы управления обучением.

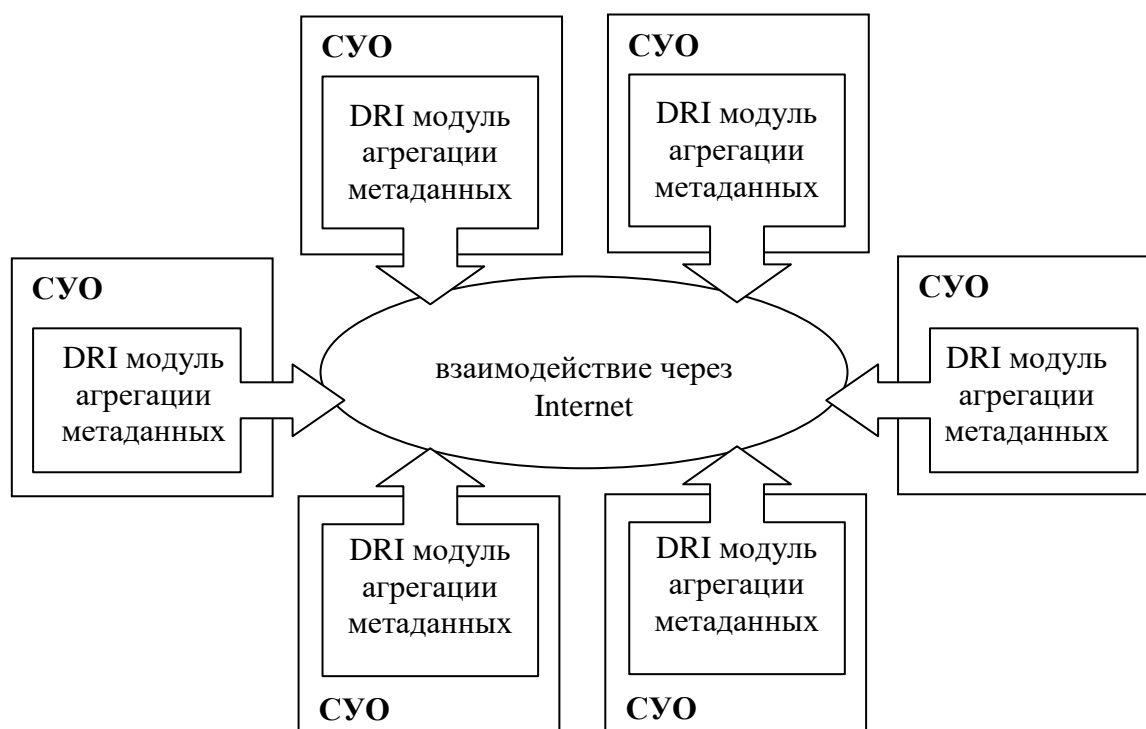


Рис. 2. Схема распределённой системы доступа к учебным ресурсам на основе существующих СУО

В рамках научно-исследовательской работы на кафедре ИКГ СПбГУ ИТМО начата разработка модулей расширения для СУО Moodle с целью практической реализации описанного предлагаемого подхода и создания открытой распределённой системы поиска учебных материалов для российских ВУЗов.

ПРИНЦИПЫ ПОСТРОЕНИЯ ЭЛЕКТРОННЫХ УЧЕБНЫХ ПОСОБИЙ ИЗ ИНФОРМАЦИОННЫХ МОДУЛЕЙ

В связи с развитием компьютерных и сетевых технологий в последние два десятилетия большую популярность получило использование в учебном процессе электронных учебных пособий. Вследствие этого за прошедшие годы появилось немало различных электронных пособий, начиная от простых текстовых электронных учебников и заканчивая сложными интерактивными обучающими программами и разнообразными системами электронного контроля знаний, умений и навыков. При этом в разных учебных пособиях можно порой встречать одни и те же содержательные элементы (например, определения каких-либо терминов, поясняющие схемы, контрольные задания и т.п.). В то же время разные учебные пособия могут иметь различное стилистическое оформление, могут быть рассчитаны на использование в разных электронных средах или с разными целями - таким образом, способ представления одной и той же информации в разных пособиях может быть разным. В результате этого возникает противоречие, требующее найти ответ на вопрос: «Как представлять информацию в электронных учебных материалах, чтобы можно было при необходимости использовать её в разных пособиях?»

Это противоречие может быть разрешено за счёт использования модульного принципа построения электронных учебных материалов.

Цель работы – исследовать принципы, по которым следует строить электронные учебные материалы, чтобы обеспечить возможность их повторного использования при разработке новых пособий.

Для достижения поставленной цели необходимо решить следующие задачи:

1. описать общую модель модульного учебного пособия;
2. определить основные принципы построения модульного учебного пособия;
3. выбрать технологические средства, наиболее подходящие для разработки модульных учебных материалов в соответствии с описанной моделью;
4. разработать программное инструментальное средство, позволяющее создавать модульные учебные пособия в соответствии с описанными принципами.

Общий анализ электронных учебных материалов позволил выделить в них такие две составляющие, как учебное содержимое (та информация, которая непосредственно описывает формируемые у учащегося знания, умения, навыки) так и параметры отображения учебного содержимого (описание того, в каком виде учебное содержимое будет представлено учащемуся).

Признаком, отличающим учебное содержимое от параметров отображения, является то, что для любого учебного содержимого можно сформулировать одну или несколько образовательных целей (формирование либо контроль конкретного знания, умения, навыка, способности, мотивации, мировоззрения и т.п.). Параметры отображения не могут иметь образовательных целей. К параметрам отображения можно отнести:

- параметры оформления (например, выделение текста жирностью или курсивом; цвет, размер и т.п.) – кроме тех случаев, когда в учебном пособии идёт изучение конкретных особенностей оформления (например, рассматривается влияние цвета на восприятие – в этом случае информация о цвете рассматриваемых объектов является учебным содержимым);

- параметры расположения содержательных элементов (координаты, размер и т.д.) – кроме случаев, аналогичных рассмотренному в предыдущем пункте;

- средства реализации интерактивности (взаимодействия элементов учебного пособия с учащимся) и интероперабельности (взаимодействия учебного пособия с системой управления обучением, в которой оно используется);

- специфические особенности формата данных, в котором представлен учебный материал.

Учебное содержимое может быть разбито на элементарные составные части, для каждой из которых можно выделить одну простейшую образовательную цель. Исходя из принципа модульности, эти элементарные составляющие можно рассматривать как учебно-информационные модули. Учебное содержимое, которое строится из нескольких таких элементарных частей, тоже может быть рассмотрено как учебно-информационный модуль, имеющий свои образовательные цели, и оно тоже может быть использовано для построения ещё более крупных модулей. Таким образом, в качестве учебно-информационных модулей можно рассматривать как совсем небольшие фрагменты учебного содержимого (например, отдельные определения терминов, поясняющие изображения и т.п.), так и более крупные (слайды обучающих программ, главы учебников, учебники целиком или даже целый учебный курс).

Параметры отображения могут описываться отдельно от учебного содержимого и применяться к отдельным элементам учебно-информационного модуля в зависимости от функционального назначения этих элементов. Таким образом, внутри модуля должны быть выделены структурные элементы, для которых указано их функциональное назначение. Логически связанные параметры отображения могут быть сгруппированы в шаблоны отображения и при необходимости применяться к разным модулям, имеющим сходную структуру, что отображено на рис. 1.

Подобное рассмотрение электронных учебных материалов было формализовано в следующей модели: электронное учебное пособие состоит из учебно-информационных модулей (УИМ), которые могут вкладываться друг в друга, и параметров отображения этих модулей.

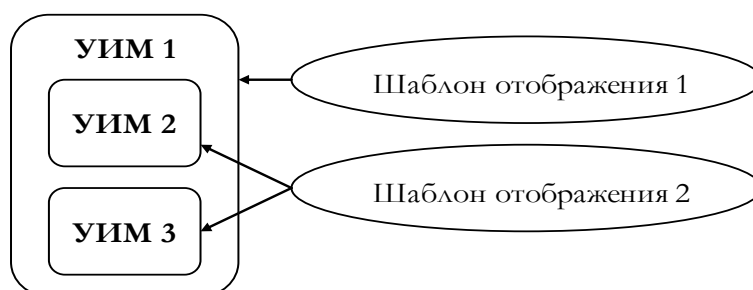


Рис. 1. Модели построения учебного материала из учебно-информационных модулей и шаблонов отображения

На основе этой модели были выведены следующие **принципы** построения модульных электронных учебных пособий:

1. Разделение учебного материала на отдельные небольшие части. В идеале, каждый содержательный элемент, который теоретически может быть использован в другом пособии, должен быть описан обособленно от остальной части учебного материала.

2. Отделение учебного содержимого от параметров его отображения, благодаря чему проще использовать одни и те же содержательные элементы в разных учебных пособиях, независимо от того, какое стилистическое оформление имеет пособие, в каком формате данных оно представляется и в каких электронных средах обучения оно используется.

3. Структурирование учебного содержимого по функциональному назначению входящих в него элементов, что позволит применять конкретные параметры отображения к разным функциональным частям учебного материала.

В качестве **средств** технической реализации электронных учебных материалов в соответствии с описанной моделью были выбраны следующие форматы и технологии:

- XML - для представления структурированного содержимого;
- XSLT – для описания шаблонов отображения;
- CSS - для описания стилового оформления;
- JavaScript - для реализации интерактивности и интероперабельности.

Выбор этих технологий объясняется тем, что они широко распространены, являются рекомендацией всемирного консорциума W3C и применяются в большинстве современных браузеров, которые являются средой выполнения электронных учебных пособий. При необходимости для обработки материалов в формате XML и XSLT могут быть применены и многие другие современные технологии, например, Flash или Java applet, т.к. в них имеются встроенные средства работы с этими форматами.

С учётом выбранных технологических средств была разработана схема формирования модульного электронного учебного материала, применение XSLT-шаблона к XML-документу даёт на выходе HTML-документ, отображаемый в браузере. Наличие JavaScript-кода в шаблоне отображения добавит интерактивность или интероперабельность в электронный документ, в результате чего его можно будет уже рассматривать как Dynamic HTML (DHTML).

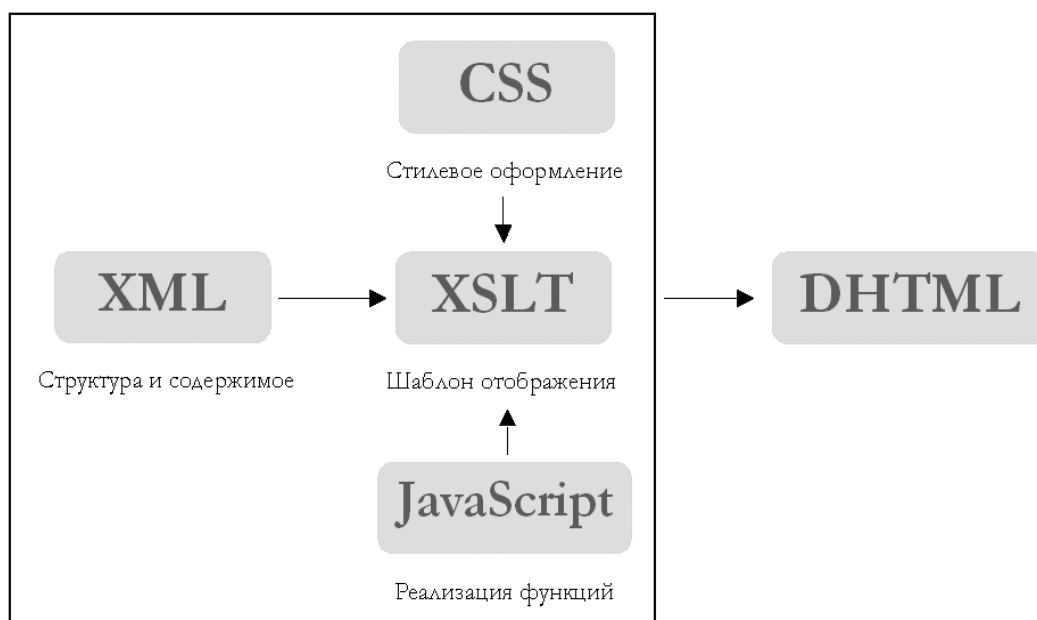


Рис. 2. Схема формирования модульного электронного учебного материала

Чтобы упростить работу преподавателя по разработке структурированных модульных электронных учебных материалов, необходимо разработать редактор, который позволит создавать и редактировать учебные материалы, не вдаваясь в технические подробности форматов и технологий. В рамках данной работы такой редактор был разработан и опробован на практике.

Итак, в работе были достигнуты следующие результаты:

- составлена общая теоретическая модель модульного учебного пособия;
- обоснованы принципы построения модульного учебного пособия;
- разработано программное инструментальное средство, позволяющее создавать модульные учебные пособия в соответствии с принципами, обеспечивающими возможность повторного использования их частей при разработке новых пособий.

СОДЕРЖАНИЕ

А.М. Заяц. Итоги научно – исследовательской работы.....	3
И.В. Панфилов. Определение технической надежности комплекса ЭВМ.....	5
В.А. Богатырев, С.В. Богатырев, А.В. Богатырев, Оптимизация разбиения кластера на подгруппы с ограниченной доступностью.....	7
В.А. Богатырев, В.Ф. Беззубов, Е.Ю. Котельникова А.А. Землянухин, С.И. Белоусов, А.С. Румянцев, А.В. Богатырев. Выбор структуры отказоустойчивого двухмашинного вычислительного комплекса	14
В.А. Богатырев, А.В.Осипов. Оценка надежности сети zigbee.....	21
С.В., Гуров. Надежность технической системы при изменении условий ее эксплуатации	26
Н.П. Васильев. Сплайн интерполяция с нетрадиционными условиями сшивки.....	39
В.А. Горбачев. Программная реализация формирования условий поиска в электронном каталоге гербария глта в среде asp.net.....	42
Джерихов В.В. Тестирование знаний студентов в среде mytest x....	52
Ю.А. Жук. Проблемы разработки и использования дисплейных форм наглядности.....	58
В.В. Илющенко. Распознавание границ дефектов на поверхности пиломатериала.....	63
А.В.Лавров. Поиск учебных материалов по метаданным.....	67
Лавров А.В. Принципы построения электронных учебных пособий из информационных модулей.....	72

Научное издание

ИНФОРМАЦИОННЫЕ СИСТЕМЫ
И ТЕХНОЛОГИИ:
ТЕОРИЯ И ПРАКТИКА

Сборник научных трудов

Выпуск 3
Часть 2

Отпечатано в авторской редакции с готового оригинал-макета

Подписано в печать с оригинал-макета
Формат 60x84/16. Бумага офсетная. Печать трафаретная.
Уч.-изд. л. 8,0. Печ. л. 8,0. Тираж 100 экз. Заказ №

Санкт-Петербургская государственная лесотехническая академия
Издательско-полиграфический отдел СПбГЛТА
194021, Санкт-Петербург, Институтский пер.,5.

