



Содержание

Редакционный совет:

В. М. Кутузов
(председатель)
М. Ю. Шестопалов
(заместитель председателя)
В. А. Мейев
(ответственный секретарь)
В. Н. Малышев
(«Радиоэлектроника
и телекоммуникации»)
А. В. Соломонов
(«Физика твердого тела
и электроника»)
М. С. Куприянов
(«Информатика, управление
и компьютерные технологии»)
В. В. Путов
(«Автоматизация и управление»)
Г. И. Прокофьев
(«Электротехника»)
Ю. В. Филатов
(«Приборостроение
и информационные
технологии»)
В. В. Шаповалов
(«Биотехнические системы
в медицине и экологии»)
С. А. Степанов
(«Управление качеством,
инновационный
и антикризисный менеджмент»)
О. Ю. Маркова
(«Гуманитарные науки»)
О. Г. Вендик
(«История науки,
образования и техники»)
Н. В. Лысенко
(«Современные технологии
в образовании»)

197376, Санкт-Петербург,
ул. Проф. Попова, 5
СПбГЭТУ «ЛЭТИ»
Тел.: (812) 234-02-23

КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ В ОБРАБОТКЕ СИГНАЛОВ

- Герцман В. Е., Суворов А. А., Сигов Е. Н.* Задача идентификации источников лазерного облучения навигационных космических аппаратов 3
- Кулаков С. А., Разумовский Г. В.* Система учета проезда железнодорожных вагонов 8
- Горошков И. А., Кулаков С. А.* Детектор распознавания присутствия людей в кадре 13
- Орешко Н. И.* Восстановление закона изменения гетероскедастического шума при траекторных измерениях на основе вейвлет-технологии 16
- Голяк С. С., Гордеев С. А., Горошков И. А., Разумовский Г. В.* Комплексная система видеонаблюдения за технологическими процессами 21
- Жукова Н. А., Соколов И. С., Экало А. В.* Метод нечеткого сравнения телеметрических параметров на основе алгоритмов интеллектуального анализа данных 25
- Фирсов М. А., Ивановский С. А.* Параллельная реализация алгоритма построения пересечения простых полигонов с использованием технологии CUDA 29

ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

- Литвиненко Н. М., Родионов С. В.* Универсальный конструктор структур данных для систем каталогизации информации 35
- Кирьянчиков В. А., Романов А. С.* Автоматическая генерация тестов объектно-ориентированных программ по диаграммам деятельности 38
- Калишченко Е. Л., Романенко Ю. В.* Технологические аспекты реализации компьютерных обучающих систем с трехмерной графикой в среде QNX 43
- Романенко С. А., Скрипникова А. С.* Управление процессом разработки на основании требований 51
- Черная О. С., Федорова Ю. Ю., Беляев С. А.* Применение методов и средств автоматизированного тестирования для проверки качества программных комплексов обработки измерительной информации 55
- Куприянов М. С., Шичкина Ю. А.* Схема укрупнения операций распараллеливаемого алгоритма 58

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В УПРАВЛЕНИИ СЛОЖНЫМИ СИСТЕМАМИ

- Васильев А. В., Жукова Н. А.* Методы оценивания информативности результатов опытной эксплуатации сложных многопараметрических объектов в условиях неопределенности 63
- Лисс А. А., Скрипникова А. С.* Автоматизированная система архивирования и аудита электронного обмена в рабочих процессах таможенного дела 67
- Холод И. И.* Применение методов Data Mining для оценки выполнения программных мероприятий предприятиями ОПК 70

<i>Холод И. И., Родионов С. В.</i> Построение централизованных хранилищ данных для систем управления предприятиями ОПК	73
<i>Беляев С. А.</i> Реализация положений ITIL/ITSM в подсистеме управления и контроля функционирования ГАС «Выборы».....	76
<i>Романенко С. А.</i> Внедрение системы проектного управления как инструмент повышения эффективности деятельности современной IT-компании.....	79
<i>Шестопалов М. Ю.</i> Системный подход к построению математических моделей объектов отказоустойчивого управления.....	86
<i>Королёв Ю. А., Суворов А. А., Герцман В. Е., Зименков И. В.</i> Проблемы реализации дистанционного управления измерительными средствами	89

КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ ПРОФЕССИОНАЛЬНОЙ ПОДГОТОВКИ СПЕЦИАЛИСТОВ

<i>Романенко Ю. В.</i> Технология разработки компьютерных обучающих средств для подготовки эксплуатирующего персонала космического ракетного комплекса	96
<i>Пантелеев М. Г., Рюмин А. В., Лебедев С. В.</i> Программный комплекс моделирования пусков для подготовки специалистов-анализаторов в области ракетной техники	101
<i>Сведения об авторах</i>	107
<i>Правила представления рукописей авторами</i>	110

ПОДПИСНОЙ ИНДЕКС 45821 ПО ОБЪЕДИНЕННОМУ КАТАЛОГУ «ПРЕССА РОССИИ». ТОМ 1 «ГАЗЕТЫ И ЖУРНАЛЫ»

Подписка производится в любом почтовом отделении России

Журнал входит в Перечень ведущих рецензируемых научных журналов и изданий, в которых должны быть опубликованы основные научные результаты диссертаций на соискание ученых степеней доктора и кандидата наук (решение Президиума Высшей аттестационной комиссии Минобрнауки России от 19 февраля 2010 года № 6/6)

Учредитель:

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Санкт-Петербургский государственный электротехнический университет „ЛЭТИ“ им. В. И. Ульянова (Ленина)» (СПбГЭТУ)
197376, Санкт-Петербург,
ул. Проф. Попова, 5
Тел.: (812) 346-44-87
Факс.: (812) 346-27-58

Свидетельство о регистрации

ПИ № ФС2-8390
от 04.12.2006 г. выдано
Управлением Федеральной службы по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия по Северо-Западному федеральному округу

Редакторы:

Э. К. Долгатов,
Н. В. Лукина, О. Р. Крумина

Комп. верстка:

А. И. Рыбца, А. С. Судовцова

Подписано в печать 15.11.13.

Формат 60 × 84 1/8.

Бумага офсетная. Печать офсетная.

Гарнитура «Times New Roman».

Уч.-изд. л. 10,0. Печ. л. 14,0.

Тираж 300 экз. (1-й завод 1–120 экз.)

Заказ 125.

Издательство СПбГЭТУ «ЛЭТИ»

197376, Санкт-Петербург,
ул. Проф. Попова, 5
Тел.: (812) 346-45-23
Тел. / факс: (812) 346-28-56



УДК 629.78

В. Е. Герцман, А. А. Суворов, Е. Н. Сигов

Задача идентификации источников лазерного облучения навигационных космических аппаратов

Рассматриваются математические методы, позволяющие по частоте регистрации электромагнитных импульсов определить факт целенаправленного облучения космического аппарата лазерными импульсами и оценить географические координаты наземного источника облучения.

Квантово-оптическая станция, электромагнитный импульс, лазерное излучение, определение местоположения источников излучения, ГЛОНАСС

Стандартная навигационная задача, решаемая с использованием глобальных космических навигационных систем ГЛОНАСС/GPS, заключается в определении координат и времени потребителя, принимающего навигационные сигналы от спутников с известными координатами и бортовыми шкалами времени, синхронизированными с единой опорной шкалой.

Известна инверсия этой навигационной задачи, когда определяются координаты наземного источника и время излучения по засечкам времен приема одного излученного импульса несколькими навигационными спутниками.

В статье рассматривается малоизвестный вариант навигационной задачи, а именно определение координат наземного источника излучения в условиях, когда указанный источник на любом интервале времени облучает только один из всей группировки навигационных спутников.

Под наземным источником облучения подразумевается квантово-оптическая станция (КОС), облучающая спутник лазерными импульсами с постоянным периодом повторения (в диапазоне от 10^{-2} до 10 с) на любом из сеансов облучения. Сеанс облучения ограничен: от единиц до десятков минут. За длительный интервал времени от нескольких часов до нескольких суток может набраться значительное число сеансов облучения, проведенных данной КОС как по одному, так и по нескольким спутникам космической группировки ГЛОНАСС.

Бортовая аппаратура навигационного космического аппарата (НКА) фиксирует моменты приема электромагнитных импульсов, как от неизвестной КОС, так и от других мешающих источников естественного происхождения. Указанные моменты, привязанные к системной шкале времени ГЛОНАСС, в виде специальных информационных сообщений (СИС) передаются в наземный центр обработки информации.

В результате апостериорной обработки необходимо идентифицировать факт несанкционированного облучения с привязкой к дате и времени регистрации, определить период повторения импульсов лазерного излучения и географические координаты КОС.

Основными предпосылками для решения задачи определения координат неизвестной КОС являются:

- использование эффекта Доплера при обработке моментов регистрации импульсов в предположении, что период излучения импульсов наземной КОС строго постоянен в течение сеанса измерений;

- прецизионная система временной привязки импульсов при высокой стабильности бортовой шкалы времени (ШВ) НКА в течение сеанса измерений;

- наличие большого числа сеансов за длительный период времени по НКА, принадлежащим к разным орбитам космической группировки ГЛОНАСС.

Анализ и предварительная обработка специальных информационных сообщений. Анализ СИС с целью регистрации фактов работы КОС проводится по следующим признакам:

- регулярный характер поступления сигналов по периодичности интервалов между моментами регистрации;
- повторяемость сеансов регистрации сигналов по отношению к положению данного НКА на орбите;
- повторяемость сеансов регистрации сигналов для различных НКА по отношению к географическому району, над которым находится спутник.

По полученным признакам формируются устойчивые критерии и проводится статистический анализ гипотез о целенаправленной работе наземной КОС по аппаратам космической группировки, при этом оцениваются дата и время регистрации, частота и интервал облучения, вероятности ложной тревоги и пропуска события.

В СИС содержится следующая информация, необходимая для идентификации фактов работы наземных КОС:

- системный номер НКА (номер точки);
- дата в формате DD MM YYYY (день, месяц, год);
- время в формате hh mm ss (часы, минуты, секунды);
- номер такта регистрации N относительно целой секунды при известной длительности такта δ (микросекунды).

Для анализа СИС, поступивших в разные даты, время регистрации импульсов переводится в секунды от опорного времени и даты

$$t = \text{datetosec}(\text{DD MM YYYY hh mm ss}) + N\delta \cdot 10^{-6},$$

где datetosec – функция приведения даты и времени к числу секунд, прошедших от полуночи 5/6.01.80.

В качестве математической модели для анализа СИС примем смесь двух процессов:

- случайного поступления импульсов от внешних возмущений природного характера, при котором время поступления сигналов подчиняется закону распределения Пуассона;
- детерминированного облучения НКА наземной КОС, при котором сигналы поступают с определенной тактовой частотой с возможным кратным пропуском тактов.

Для проверки адекватности предлагаемой модели были рассмотрены несколько реальных выборок СИС с двух групп НКА ГЛОНАСС на сеансе наблюдения длительностью несколько суток:

- НКА, которые заведомо не подвергались воздействию лазерного облучения КОС;
- НКА, которые подвергались воздействию лазерного облучения КОС с известными характеристиками.

Анализ реальных СИС, поступивших от НКА, которые не подвергались облучению, показывает, что процесс случайного поступления импульсов от внешних возмущений можно считать пуассоновским со средней плотностью числа событий за единицу времени $\alpha = (5 \dots 10)E-05$ и рассчитывать вероятности поступления случайных импульсов по следующим формулам [1]:

$$p_0(T) = e^{-\alpha T}, p_i(T) = e^{-\alpha T} \frac{(\alpha T)^i}{(i)!},$$

$$p_{k \geq 2}(T) = 1 - p_0(T) - p_1(T),$$

где T – интервал времени, в течение которого регистрируются события; $p_i(T)$ – вероятность того, что на интервале длительностью T зарегистрировано ровно i импульсов; $p_{k \geq 2}(T)$ – вероятность того, что на интервале длительностью T зарегистрировано не менее двух импульсов.

Результаты расчетов показывают, что вероятность ложной тревоги для значений параметра α , близких к полученным опытным данным, при отслеживании появления двух и более импульсов подряд на интервале до 600 с не превосходит $2E-03$.

Вероятность пропуска импульсов лазерного облучения целиком зависит от частоты работы неизвестной КОС. Анализ характеристик существующих наземных КОС показывает, что частота работы лежит в диапазоне от 0.5 до 100 Гц, поэтому вероятность пропуска 10 и более лазерных импульсов на скользящем интервале длительностью от 60 до 120 с близка к нулю.

Очевидным признаком работы КОС является наличие участков со скважностью поступления импульсов, кратной некоторой константе.

Методика идентификации сигналов неизвестных КОС строится следующим образом:

1. На скользящем интервале длительностью $T = 100$ с подсчитывается число поступивших импульсов imp :

$$\tau_1 = t_0 + (i-1)T, \tau_2 = \tau_1 + T (i=1, 2, \dots),$$

$$\text{imp}(i) = \text{find}(\tau_1 \leq t < \tau_2),$$

где i – номер интервала; t – время регистрации импульса; find – процедура подсчета числа импульсов.

2. Если число поступивших импульсов не меньше заданного значения $k_{gm} = 10$, то строится гистограмма для скажностей dt и определяются значения dt , соответствующие пяти старшим членам гистограммы:

$$\begin{aligned} dt &= \text{diff}(t), t = [t_1, t_2, \dots, t_n] \in [\tau_1, \tau_2), \\ x &= [0 : \frac{\min(dt)}{4} : \max(dt)], \\ p &= \text{hist}(dt, x), \\ [p_{\text{sort}}, x_{\text{sort}}] &= \text{sort}(p), \\ y &= x_{\text{sort}}(\max : (\max - 4)), \end{aligned}$$

где diff – процедура получения последовательных разностей; hist – процедура построения гистограммы массива dt на сетке x ; sort – процедура сортировки массива p по возрастанию с соответствующей сортировкой ячеек сетки x ; y – значения массива dt , соответствующие пяти старшим членам гистограммы.

Здесь и далее стандартные процедуры обработки данных соответствуют языку технических расчетов MatLab® [2], жирным шрифтом обозначаются массивы.

3. Если значение скажности, соответствующее максимуму гистограммы, является наименьшим общим делителем для последующих трех или четырех (по мере убывания гистограмм) значений скажности, то интервал соответствует участку работы наземной КОС и частота облучения оценивается следующим образом:

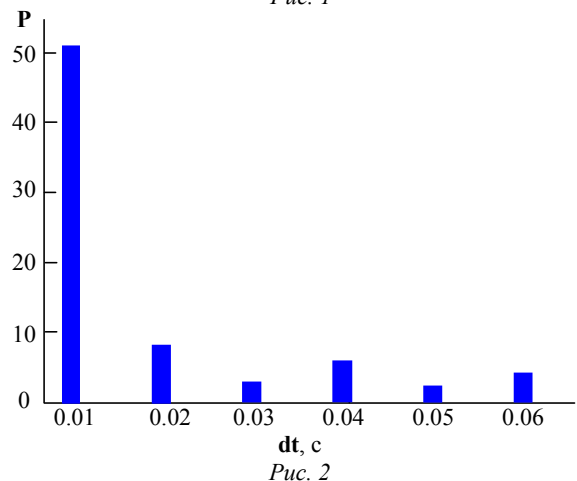
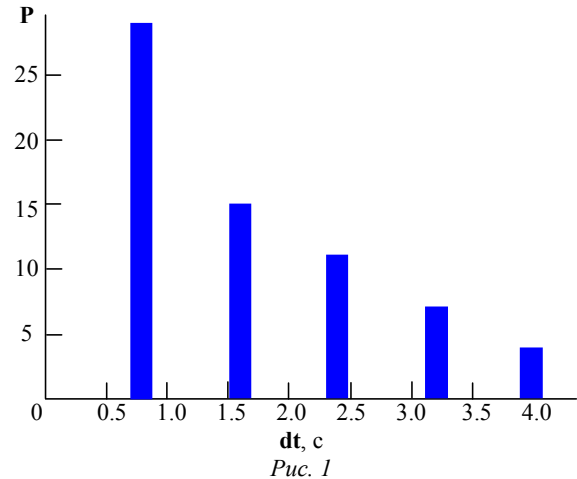
$$\begin{aligned} m &= \text{rem}(y(2:5), y(1)), \\ f_{\text{kos}} &= \frac{1}{y(1)} \end{aligned} \left. \begin{aligned} T_{\text{begin}} &= \text{sectodate}(\tau_1) \\ T_{\text{end}} &= \text{sectodate}(\tau_2) \end{aligned} \right\} \text{если } m(1:4) = 0,$$

где $\text{rem}(a, b)$ – процедура нахождения остатка от деления a на b ; f_{kos} – частота облучения КОС; $T_{\text{begin}}, T_{\text{end}}$ – границы интервала работы КОС; $\text{sectodat}(\text{sec})$ – функция приведения числа секунд, прошедших от полуночи 5/6.01.80, к текущей дате и времени.

Результаты расчетов по предлагаемой методике для сеансов измерений, соответствующих реально зарегистрированным импульсам по двум отечественным наземным КОС, приведены на рис. 1 и 2, из которых видно:

– на рис. 1 скажность принятых импульсов группируется на значениях 0.8, 1.6, 2.4, 3.2, 4.0, что соответствует частоте работы КОС 1.25 Гц;

– на рис. 2 скажность принятых импульсов группируется на значениях 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, что соответствует частоте работы КОС 100 Гц.



Единственной информацией для определения географических координат КОС, содержащейся в СИС, является изменяющийся сдвиг τ длительности интервала между моментами регистрации импульсов dt относительно номинального значения, кратного периоду излучения КОС: $T_{\text{nom}} = 1/f_{\text{kos}}$. Указанный сдвиг является аналогом эффекта Доплера вследствие изменения скорости движения спутника относительно КОС и оценивается следующим образом:

$$\begin{aligned} y &= \text{rem}(dt, T_{\text{nom}}), \\ \tau &= \begin{cases} y - T_{\text{nom}}, & \text{if } |y - T_{\text{nom}}| < \varepsilon, \\ y, & \text{if } |y| < \varepsilon, \end{cases} \end{aligned}$$

где ε – максимально возможный сдвиг при движении спутника относительно неподвижной точки на поверхности Земли.

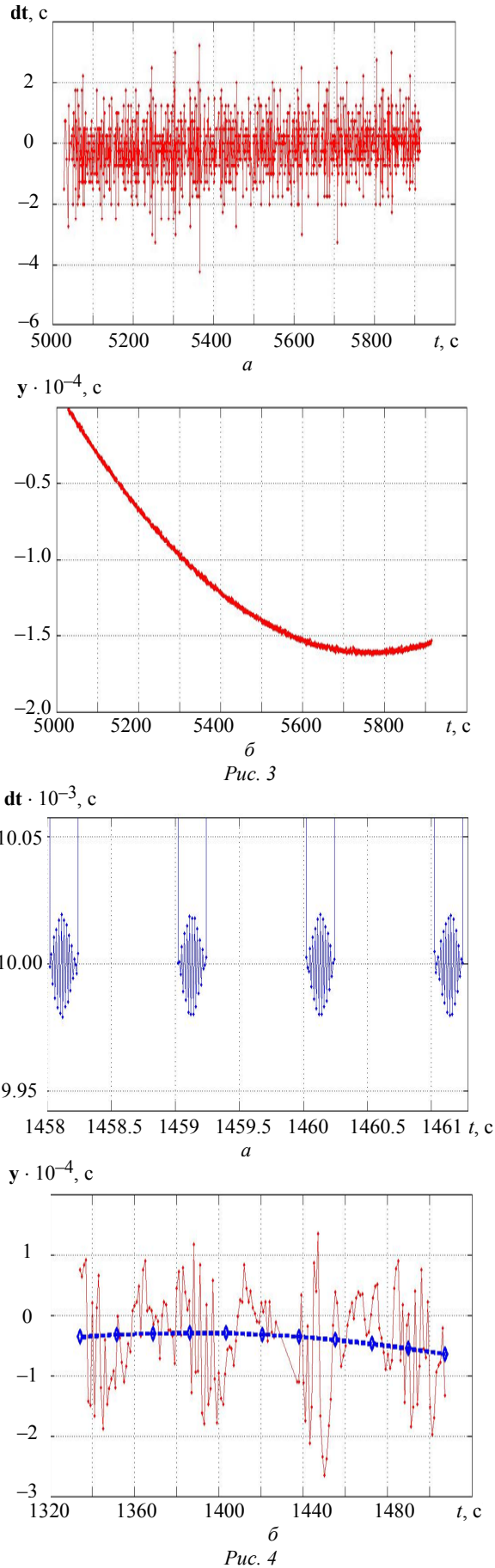


Рис. 3

На рис. 3 приведены графики изменения во времени величин dt (рис. 3, а) и y (рис. 3, б), полученные по реальным СИС для КОС с частотой излучения импульсов 1.25 Гц. Импульсы от этой КОС идут практически непрерывно на интервале порядка 15 мин. На рис. 4 приведены аналогичные графики для КОС с частотой излучения импульсов 100 Гц. Импульсы от этой КОС идут пачками на интервале порядка 3 мин.

Определение географических координат неизвестных КОС. По результатам предварительной обработки СИС определяется время регистрации на борту НКА импульсов от наземной КОС $t = [t_1, t_2, \dots, t_n]$ и приращение номинального периода следования импульсов за счет скорости движения НКА (аналог интегрального доплеровского сдвига) $\tau = [\tau_1, \tau_2, \dots, \tau_{n-1}]$. Уравнения, связывающие эти величины с координатами навигационного спутника и координатами наземной КОС, имеют вид

$$\tau_k = \frac{\Delta D_k(x_0, y_0, z_0)}{c} + 2\varepsilon_\tau,$$

$$k = (i-1)(N_{i-1}-1) + j-1 \quad (j=2, \dots, N_i, i=1, 2, \dots, M),$$

$$\Delta D_k(x_0, y_0, z_0) = |\mathbf{R}_0 - \mathbf{R}_{sv}(t_{ij})| - |\mathbf{R}_0 - \mathbf{R}_{sv}(t_{i1})|,$$

$$\mathbf{R}_0 = \|x_0 \ y_0 \ z_0\|, \mathbf{R}_{sv} = \|x_{sv} \ y_{sv} \ z_{sv}\|, \quad (1)$$

где i – номер сеанса измерений; M – число сеансов измерений; j – номер измерения внутри сеанса; N_i – число измерений внутри сеанса; ε_τ – шумовая погрешность определения момента регистрации импульса; $x_{sv} \ y_{sv} \ z_{sv}$ – известные координаты спутника в геоцентрической относительной системе координат [3] (в расчетах использовались высокоточные апостериорные эфемериды в формате SP3*); x_0, y_0, z_0 – неизвестные координаты КОС в той же системе; $|\mathbf{R}|$ – модуль вектора.

Координаты КОС определяются решением нелинейной системы уравнений (1) методом последовательных приближений. В каждом приближении система уравнений линеаризуется относительно решения, полученного на предыдущем приближении. В качестве нулевого приближения выбираются координаты центра возможного региона, в котором может быть расположена неизвестная КОС. Приближения считаются до тех пор, пока модуль разности оценок координат КОС, полученных на последовательных шагах, не становится меньше заданного значения:

* The Extended Standard Product 3 Orbit Format, <http://www.ngs.noaa.gov/orbits/sp3c.txt>.

$$\hat{\mathbf{R}}_0^p = \hat{\mathbf{R}}_0^{p-1} + \mathbf{K}_p \cdot \mathbf{Y}_p, p = 1, 2, \dots,$$

$$\mathbf{K}_p = \text{inv}\{\mathbf{H}_{p-1}\}, \mathbf{H}_p = \mathbf{H}(\hat{\mathbf{R}}_0^p),$$

$$\mathbf{Y}_p = \boldsymbol{\tau} \cdot c - \Delta D_{p-1},$$

где p – номер шага приближений; inv – процедура псевдообращения матрицы;

$$\mathbf{H}_p = \frac{\partial}{\partial x_0, y_0, z_0} \Delta D_k = \begin{bmatrix} \mathbf{N}(t_{12}) - \mathbf{N}(t_{11}) \\ \dots \dots \dots \mathbf{N}(t_{1N_i}) - \mathbf{N}(t_{11}) \\ \dots \dots \dots \mathbf{N}(t_{M2}) - \mathbf{N}(t_{M1}) \\ \dots \dots \dots \mathbf{N}(t_{MN_m}) - \mathbf{N}(t_{M1}) \end{bmatrix}$$

– матрица частных производных (первый индекс – номер сеанса, второй индекс – номер точки внутри сеанса);

$$\mathbf{N} = \frac{\mathbf{R}_{sv} - \hat{\mathbf{R}}_0^p}{|\mathbf{R}_{sv} - \hat{\mathbf{R}}_0^p|}$$

– орт направления от КОС на НКА; $\Delta D_p = |\hat{\mathbf{R}}_0^p - \mathbf{R}_{sv}(t_{ij})| - |\hat{\mathbf{R}}_0^p - \mathbf{R}_{sv}(t_{i1})|$ – вектор расчетных значений приращения периода следования зарегистрированных импульсов.

Сходимость оценки координат в методе последовательных приближений, а также погрешности указанной оценки определяются топологией двумерной поверхности, характеризующей квадратичный критерий отличия прогнозируемых значений доплеровского сдвига от измеренных:

$$R2(\varphi, \lambda) = |\boldsymbol{\tau} \cdot c - \{\mathbf{R}(\varphi, \lambda) - \mathbf{R}_{sv}(t_{ij})\} - \mathbf{R}(\varphi, \lambda) - \mathbf{R}_{sv}(t_{i1})|^2.$$

Минимум указанного критерия соответствует искомой оценке широты и долготы $\varphi_{\text{kos}}, \lambda_{\text{kos}}$ источника излучений.

На рис. 5 приведен вид двумерной поверхности $R2(\varphi, \lambda)$ над плоскостью «широта\долгота» (latitude\longitude). Сетка по широте и по долготе представлена в линейной мере в соответствии со следующими формулами:

$$B = (\varphi_{\text{kos}} - \varphi) R_{\text{earth}},$$

$$L = (\lambda_{\text{kos}} - \lambda) \cos(\varphi_{\text{kos}}) R_{\text{earth}},$$

где R_{earth} – радиус Земли, км; φ, λ – широта и долгота, рад.

Минимум квадратичной поверхности $R2(\varphi, \lambda)$ находится над точкой (0,0). Эта точка соответствует координатам КОС, для которой были обработаны результаты шести сеансов измерений при длительности каждого порядка 15 мин. Распределение сеансов по НКА для разных орбитальных плоскостей следующее: 4 сеанса – НКА третьей плоскости, 1 сеанс – НКА первой и 1 сеанс – НКА второй плоскостей. Этим объясняется несимметричность поверхности $R2(\varphi, \lambda)$. Расчеты координат КОС по измерениям, полученным в этих сеансах, показали отличия от действительных координат станции в пределах 15 км.

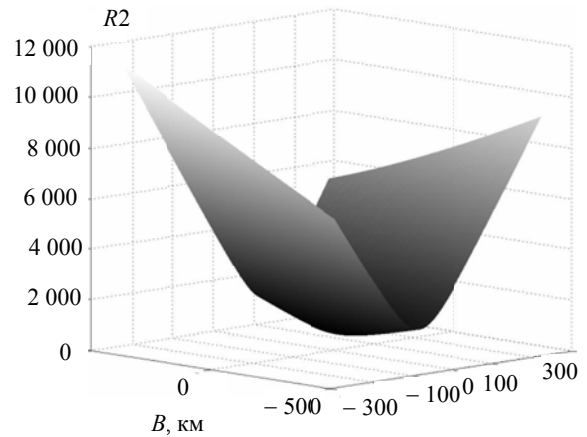


Рис. 5

Таким образом, представлена методика, позволяющая идентифицировать факт несанкционированного облучения НКА и определить на фоне естественных помех моменты регистрации лазерных импульсов, частоту работы неизвестной квантово-оптической станции и ее географические координаты.

Результаты обработки данных, полученных в реальных сеансах работы КОС по НКА ГЛОНАСС, подтвердили эффективность созданных алгоритмов. Разработанные алгоритмы реализованы в виде программных функций MatLab® и в виде полнофункционального программного комплекса на языке C++.

СПИСОК ЛИТЕРАТУРЫ

1. Корн Г., Корн Т. Справочник по математике для научных работников и инженеров. М.: Наука, 1970.
2. Иглин С. Математические расчеты на базе MatLab. СПб.: БХВ-Петербург, 2005.
3. ГЛОНАСС. Интерфейсный контрольный документ. М., 2008.

V. E. Hertzman, A. A. Suvorov, E. N. Sigov

THE PROBLEM OF LASER SOURCES IDENTIFICATION THAT IRRADIATE NAVIGATION SPACECRAFTS

The mathematical methods which allows to determine the fact of targeted laser irradiation are considered. The algorithm of the geographic coordinates estimation for ground-based illumination source is proposed

Quantum-optical station, electromagnetic pulse, laser irradiation, irradiation source position determination, GLONASS

УДК 004.932.2, 004.67

С. А. Кулаков, Г. В. Разумовский

Система учета проезда железнодорожных вагонов

Приводится описание системы распознавания номеров вагонов, ее состав и описание отдельных алгоритмов. Обсуждаются преимущества применения алгоритма определения границ вагонов в системе распознавания номеров вагонов. Отмечается возможность повышения вероятности распознавания номеров вагонов за счет применения совокупности алгоритмов, основанных на получении косвенных данных из дополнительных систем.

Алгоритм распознавания номеров вагонов, определение границ вагонов, натурный лист, трекинг

В настоящее время в области компьютерного зрения одной из актуальных задач является автоматизированное распознавание номеров железнодорожных (ж/д) вагонов, контейнеров, автомобилей и т. п. Необходимость таких систем обусловлена сложностью «ручного» визуального контроля за процессом перевозки грузов, а также их автоматизированного учета в системах логистики и управления технологическими процессами. Однако существует ряд факторов, усложняющих создание таких систем. Среди них можно выделить: возможность нефиксированного положения распознаваемых символов, их различный размер и цвет в пределах одного изображения; наличие помех, мешающих выделению символов на изображении (например, цемент, просыпанный на номер вагона, или подтеки нефти на номерах цистерн и т. п.); расположение объекта на плохо освещенном участке считывания номеров.

В случае распознавания номеров ж/д вагонов следует учитывать, что кроме основного номера на вагоне может располагаться еще и контрольный номер, нанесенный существенно меньшим шрифтом и расположенный в его нижней части,

причем его цвет может отличаться от цвета основного номера. Таким образом, в пределах одного изображения символы, требующие распознавания, могут быть изображены существенно отличающимися друг от друга цветами. Ситуация может осложняться неоднородностью освещения при съемке вагона, что приводит к изменению цвета как в пределах одной последовательности символов (номера), так и в пределах одного символа.

Традиционно задача определения номеров ж/д вагонов решается в несколько этапов:

1. Обнаружение состава и разбиение его на вагоны.
 2. Группировка набора кадров, относящихся к одному и тому же вагону.
 3. Локализация номера в кадре.
 4. Распознавание номера.
 5. Проверка правильности распознанного номера подсчетом его контрольной суммы и, по возможности, сравнением номера с данными натурального листа.
 6. Фиксация события проезда вагона в базе данных.
-

Задачу обнаружения состава и разбиения его на вагоны можно решать при помощи системы видеонаблюдения. При этом есть ряд недостатков, существенно сказывающихся на точности результата. Факторами, влияющими на качество результата, являются любые движения вдоль ж/д полотна, динамический задний фон. Поэтому для повышения точности обнаружения состава и разбиения на вагоны данная задача решается одним из нескольких способов:

- *Использование фоновой щита.* Для решения данной задачи используются фоновые щиты, расположенные напротив каждой камеры. Если состава нет, то фоновый щит виден целиком. Как только часть фоновой щита перекрывается поездом, система начинает запись кадров проходящего поезда. При этом первые кадры обрабатываются особо. На этом этапе определяется наличие поезда и направление его движения, отсеиваются различные помехи, которые могут частично закрывать фоновый щит. При обработке первых кадров выявляются также случаи, когда поезд состоит из одного локомотива. Обработка такого состава не выполняется. Недостатком данного метода является плохое распознавание щита при неблагоприятных погодных условиях или в темное время суток. Также существуют места, где невозможно установить щит.

- *Использование инфракрасных (ИК) датчиков.* Для разбиения состава на вагоны используются ИК-датчики в количестве от 1 до 3. Если используется только один датчик, то направление движения определяется при анализе изображения. При использовании нескольких датчиков направление движения определяется по последовательности срабатывания датчиков. Так как требуется определять как вагоны, так и ж/д платформы, то излучатель и приемник каждой пары ИК-датчиков необходимо размещать на разной высоте, чтобы ИК-луч проходил под углом к горизонту. К недостаткам данного метода относится частая необходимость в очистке датчиков, особенно в зимний период, и ложные срабатывания в плохую погоду (интенсивный дождь или снегопад).

- *Использование лазерного дальномера.* Достоинством такого метода, по сравнению с ИК-датчиками, являются простота использования и надежность работы. Отсутствие второго датчика (приемника) исключает установку дополнительной опоры с противоположной стороны рельсов и

прокладку кабелей ко второй опоре. Кроме того затраты на обслуживание системы разбиения на вагоны с использованием лазерного дальномера значительно ниже, так как установка датчика предусматривается на высоте не ниже 2 м, что приводит к меньшему загрязнению оптической системы и уменьшению затрат на обслуживание системы. К недостаткам использования датчиков подобного типа относятся: небольшой диапазон контролируемых расстояний (до 10 м) при умеренной мощности лазера, невысокое быстродействие при резком изменении расстояний (пересечении луча), отсутствие возможности контроля для поверхностей с поглощением света свыше 85 %, более высокая стоимость по сравнению с ИК-датчиками.

- *Использование индукционных датчиков или магнитных педалей.* Индукционные датчики имеют следующие достоинства:

- качество их работы не зависит от метеословий и времени суток;
- кроме определения границ вагона позволяют получить скорость состава и направление его движения.

Недостатками использования индукционных датчиков являются высокая стоимость оборудования и большой объем монтажных работ при их установке.

В качестве индукционных датчиков или магнитных педалей используется несколько видов датчиков, закрепленных на рельсовых путях:

1. Магнитная педаль ПБМ-56.
2. Индуктивные датчики колесных пар фирмы Siemens (WSR, WSD).
3. Индуктивные датчики колесных пар компании «Промэлектроника» ДКУ ЕРКФ.6652.002-02.
4. Индуктивный датчик Tiefenbach 2N59-1R-200-45.

Выбор датчика зависит прежде всего от скорости движения состава. При скоростях состава выше 30 км/ч необходимо использовать индуктивные датчики фирм Siemens и Tiefenbach, при меньших скоростях – датчик компании «Промэлектроника», у которого период опроса датчика не может быть менее 10 мс. Обычно номера вагонов считываются при въезде или выезде с сортировки или предприятия, где скорости движения состава небольшие. Поэтому с точки зрения массовости применения и простоты реализации в статье рассматривается использование датчика

счета колесных пар ДКУ ЕРКФ.6652.002-02. Одним из преимуществ этого датчика является наличие интерфейса RS-485, что позволяет подключать его к промышленным микроконтроллерам. К его достоинствам также можно отнести наличие двух индуктивных катушек в корпусе датчика, что дает возможность использовать один датчик для определения скорости колесной пары и направления движения вагона, и наличие встроенной электроники, позволяющей передавать на выход большой объем информации, начиная от скорости состава и заканчивая проверкой работоспособности датчика.

Задача группировки набора кадров, относящихся к одному и тому же вагону, решается на этапе анализа типа и границ вагона. Группировка начинается при проходе первой колесной пары вагона над датчиком ДКУ, а заканчивается при проходе последней колесной пары вагона. При этом все кадры, полученные с видеокамеры в этот промежуток времени, помечаются как кадры, относящиеся к одному вагону.

Задачу локализации номера в кадре и распознавания на сегодняшний день решают многими способами, среди которых наибольшую популярность имеют гистограммные методы, метод определения локального контраста, морфологические методы обработки изображения [1]. Также существуют различные методы предобработки изображения, позволяющие повысить качество распознавания номеров. В данной статье рассматривается комбинация методов локального контраста с гистограммным методом, что позволяет увеличить вероятность правильной локализации номера на кадре.

Задача распознавания символов номера вагона может быть решена одним из существующих на данный момент методов [2]: сравнением с эталонными шаблонами, контурным анализом (скелетизация) и использованием нейронных сетей. Вероятность распознавания номеров ж/д вагонов данными способами достигает 80...90 % на всем парке ж/д вагонов. Важным критерием, обеспечивающим высокую достоверность распознавания номера вагона, являются характеристики видеокамеры. Среди них следует особо выделить скорость затвора срабатывания камеры. Скорость затвора должна быть достаточно высокой при любом освещении, так как вагон движется пер-

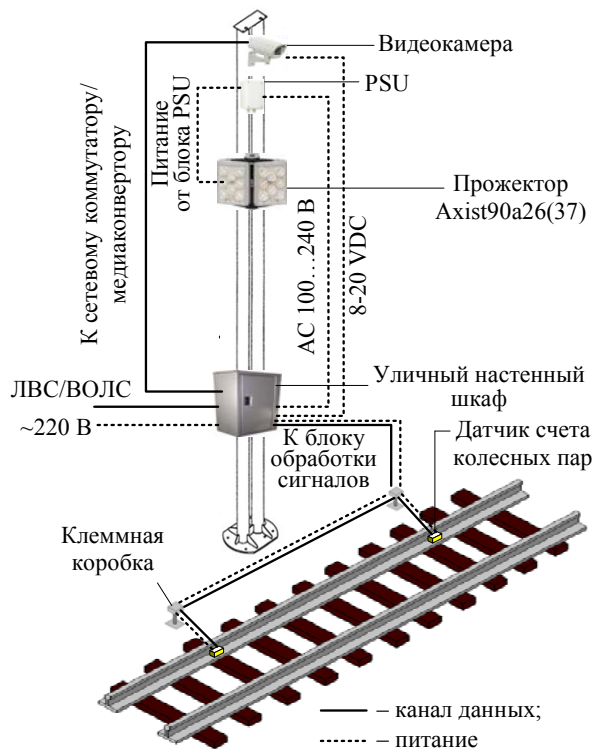
пендикулярно оптической оси камеры и при недостаточной скорости затвора изображение будет размытым, что приведет к невозможности распознавания номера.

Наибольший эффект дает метод распознавания номеров ж/д вагонов, основанный на получении дополнительной информации о типе вагона еще на этапе формирования последовательности кадров, относящейся к конкретной единице подвижного состава (ЕПС) вагона. Все вагоны классифицируются по типам, и априорное знание одной или нескольких цифр номера вагона позволяет повысить вероятность правильного распознавания номера.

Грузовые вагоны обращаются по всей сети железных дорог и имеют нумерацию, построенную по специальной системе. Номер каждого грузового вагона состоит из восьми цифр. В этой комбинации в неявном виде присутствует осьность вагона и последняя цифра, полученная из комбинации предыдущих семи, является контрольной в соответствии со стандартом ОАО «РЖД».

Получение типа вагона возможно по определению осьности и геометрических характеристик, присущих всем единицам подвижного состава и неизменным при любых скоростях и загрузке. Эту информацию можно получить при помощи ДКУ, закрепленных на рельсовых путях, и системы обработки данных. Для определения номеров вагонов и подсчета количества единиц подвижного состава применяется трехуровневая схема обработки данных. На первом уровне обработки используется контроллер, который установлен непосредственно в пункте слежения за ж/д составом и выполняет функцию опроса ДКУ и передачу информации при изменении данных на счетчике колесных пар на следующий уровень обработки. Общий вид пункта слежения приведен на рисунке.

Модуль опроса датчиков колесных пар и передачи данных на сервер аналитики написан на языке C и реализован на микрокомпьютере MOXA UC7110-T-Lx. Данный модуль выполняет опрос датчиков с периодичностью 10 мс и в случае изменения значения на счетчике количества осей передает данные по сети Ethernet на сервер для дальнейшего анализа. Период опроса датчиков позволяет с высокой точностью определять межосевые расстояния, и погрешность составляет не более 4 см.



Второй уровень обработки принимает сообщения от микрокомпьютера с данными по каждой из осей и определяет тип и количество единиц подвижного состава, распознает номера вагонов и передает сформированные события на сервер обработки событий. Анализ всех типов подвижных единиц позволил установить, что по размерам межосевых расстояний можно определить ее тип [3]. Например, на основе измеренных расстояний между осями в тележке определяется ее тип. Далее, на основе информации о типе тележки и дополнительной информации о количестве осей, расстоянии между колесными парами разных тележек определяется конкретный тип единицы подвижного состава [4].

Система определения типов и границ вагонов работает следующим образом. После прохождения каждой колесной пары контроллер посылает данные о скорости движения колесной пары в зоне действия индуктивного датчика и времени проезда колеса. Межосевое расстояние рассчитывается усреднением скоростей движения соседних осей и умножением на разницу во времени прохождения осей. После прохождения первых трех колесных пар в зоне действия датчика и определения расстояний между ними запускается алгоритм идентификации типа единицы подвижного состава. При этом полученные межосевые

расстояния сравниваются со справочником всех типов вагонов и локомотивов, хранящихся в базе данных системы. По мере поступления с контроллера новых данных о прохождении колесных пар количество идентифицированных типов будет уменьшаться и, в конечном счете, останется только тот тип единицы подвижного состава, который соответствует расстоянию между тележками и осями вагона.

После идентификации типа вагона и в момент прохождения последней оси над датчиком завершает работу алгоритм распознавания и накопления видеоданных и запускается алгоритм формирования номера вагона. На следующем шаге происходит обнуление массива с данными о межосевых расстояниях и программа переходит в режим ожидания следующей единицы подвижного состава.

В местах распознавания номеров вагонов, где возможно маневрирование ж/д состава, требуется установка второго индуктивного датчика. Это связано с тем, что в момент остановки ж/д состава теряется информация о межосевом расстоянии и идентификация типа вагона при потере более одного из межосевых расстояний становится невозможной.

При установке дополнительного датчика алгоритм меняется следующим образом. Поскольку датчики устанавливаются на некотором расстоянии друг от друга и симметрично относительно оптической системы распознавания номеров вагонов, то запуск алгоритма распознавания номеров производится при прохождении второй оси вагона над первым ДКУ, а остановка алгоритма происходит при прохождении предпоследней оси вагона над вторым датчиком. Данные о межосевых расстояниях объединяются из двух массивов по каждому из датчиков, и «потерянные» расстояния в одном из массивов дополняются из другого. Также вводится дополнительная переменная для хранения количества осей для предыдущей единицы подвижного состава. Из второго массива данные удаляются после того, как в зоне действия второго датчика пройдет последняя ось предыдущего вагона. Затем переменная с количеством осей для предыдущего вагона обнуляется.

Также в системе существует дополнительный модуль, который автоматически запускается при сбоях различного характера. Возможны сбои в работе ДКУ, сбои в работе микроконтроллера и сбоя в работе сервера аналитики. Для обнаружения сбоя в ДКУ микроконтроллер с некоторым интервалом опрашивает датчик и в случае обна-

ружения сбоя посылает сигнал на перезагрузку датчика. Если перезагрузка не помогает, то алгоритм распознавания определяет границы вагона, используя видеоданные, поступающие от камеры. В этот же режим алгоритм переходит и при обнаружении сбоя в работе микроконтроллера. Также возможны ситуации, когда во время перезагрузки ДКУ над ним прошла колесная пара и алгоритм «потерял» возможность определения типа вагона. В этом случае по мере прохода колесных пар над ДКУ производится «поиск типа вагона» в списке межосевых расстояний.

При формировании номера вагона каждый из символов проверяется по всей выборке номеров, принадлежащей текущему вагону, и выбираются символы с наилучшей вероятностью распознавания. Сформированный номер проверяется при помощи контрольной цифры, указанной в восьмом символе номера в соответствии со стандартом ОАО «РЖД»*. Если возможен доступ к базе данных с натурными листами, то можно осуществлять контроль правильности распознавания номера вагона, сравнив его с номерами в натурных листах. Если распознанный номер совпадает с одним из номеров в натурном листе, считается, что номер распознан со 100 %-й вероятностью. Если точного соответствия номера в натурных листах не найдено, то выбирается максимально похожий номер в соответствии с вычисленным расстоянием Левенштейна между распознанным номером вагона и всеми номерами из натурного листа.

Третий уровень обработки событий осуществляет прием информации от сервера аналитики. На нем запущена программа обработки событий и находится база данных по всем событиям. Сервер аналитики отправляет данные с номером, типом, направлением движения вагона и фотографией вагона с номером.

Рассмотренная система учета железнодорожных единиц подвижного состава на основе вычисления геометрических характеристик ж/д вагонов позволяет повысить вероятность распознавания номеров железнодорожных вагонов до 96 % за счет следующих факторов:

1. Контроля первых трех цифр номера вагона сопоставлением их с цифрами, которые вычисляются на основании типа вагона.

2. Точного определения границ вагона с возможностью группировки всех кадров, относящихся к каждому вагону, что позволяет получить максимальное число кадров для анализа номера вагона и исключить кадры, в которых присутствует другой вагон.

3. Анализа номера в нескольких кадрах и многократного анализа одной и той же цифры на нескольких кадрах.

4. Использования контрольных сумм и натурных листов.

Данная система также может быть использована для подсчета единиц подвижного состава, номера на которые не нанесены или не могут быть прочитаны.

СПИСОК ЛИТЕРАТУРЫ

1. Система технического зрения для распознавания номеров железнодорожных цистерн с использованием модифицированного коррелятора в метрике Хаусдорфа / С. Г. Волотовский, Н. Л. Казанский, С. Б. Попов, Р. В. Хмелев; Ин-т систем обработки изображений РАН. 2007.

2. Волотовский С. Г., Казанский Н. Л., Попов С. Б. Система регистрации железнодорожных составов цистерн: Свидетельство об официальной регистрации программ для ЭВМ № 2004611969 по заявке

№ 2004611381 от 29 июня 2004 г. Зарегистрировано в Реестре программ для ЭВМ 26 авг. 2004 г.

3. Ромкин М. В., Тюмиков Д. К. Геометрические параметры для идентификации единиц железнодорожного подвижного состава // Вестн. транспорта Поволжья. 2008.

4. Ромкин М. В., Засов В. А. Система автоматической идентификации типов подвижных единиц железнодорожного транспорта // СамГУПС. 2010.

S. A. Kulakov, G. V. Razumovskiy

ACCOUNTING SYSTEM MAP RAIL CARS

The article describes the system of recognition of numbers of cars, its composition and the description of the individual algorithms. The advantages of applying the algorithm for determining the boundaries of cars in the system of recognition of numbers of cars. Mentions the possibility of increasing the probability of recognition of numbers of cars, due to application of a set of algorithms based on the receipt of proxy data from additional systems.

Recognition algorithm numbers of cars, the definition of the boundaries of cars, full-scale list, tracking

* http://logisticsinfo.ru/main/art_tran_provpodbor.shtml.

УДК 004.932.2

И. А. Горошков, С. А. Кулаков

Детектор распознавания присутствия людей в кадре

Приводится алгоритм распознавания присутствия человека в кадре. Обсуждаются преимущества выбранного алгоритма определения людей. Отмечается возможность применения данного алгоритма для обеспечения сохранности ТМЦ (товарно-материальные ценности) и безопасности на предприятиях любых сфер и направлений деятельности.

Алгоритм распознавания людей, HOG, машинное обучение, трекинг

Задача обнаружения людей на изображениях становится все более актуальной в области компьютерного зрения и распознавания образов. Это связано с возможностью применения алгоритмов обнаружения человека в системах видеонаблюдения для охраны объектов, контролем за техникой безопасности на промышленных предприятиях, наблюдением в торговых комплексах в целях изучения спроса и пр. Идеальный детектор должен обнаружить всех присутствующих на цифровом изображении людей независимо от их расположения, масштаба или позы. Однако автоматическое детектирование человека на снимке является довольно непростой задачей из-за ряда факторов: детектору приходится иметь дело с различным освещением, различным положением тела человека и одежды, сложным рельефом местности, частичным или полным перекрытием человека различными предметами или автомобилями. Также детектор должен отличать людей от подвижных механизмов, схожих по размеру с человеком (например, грейфер или магнит на мостовом кране). Кроме того детектор должен определять людей в большом диапазоне внутригрупповых изменений и при этом должен иметь как можно меньше ложных срабатываний на областях фона.

В связи с актуальностью и сложностью задачи детектирования фигуры человека на настоящий момент существует множество разнообразных подходов к ее решению. Так, предпринимались попытки осуществить детектирование сравнением предобработанных изображений с некоторыми эталонами (шаблонами) с помощью специальной метрики [1], использования детекторов и дескрипто-

ров особых точек; применения сегментации изображения [2] и т. д. Одним из наиболее популярных и перспективных на настоящий момент подходов к детектированию объектов на изображении является так называемый метод бегущего окна. Данный метод основан на осуществлении экстенсивного поиска объектов определенного размера. В каждом кадре последовательно рассматриваются области изображения заданного размера и им ставится в соответствие признаковое описание, на основе которого с помощью алгоритма классификации принимается решение, содержит ли данная область объект или нет. Более подробно данный подход и особенности его реализации рассматриваются далее, здесь же отметим, что ключевыми компонентами системы детектирования, основанной на таком подходе, являются алгоритмы получения признакового описания изображения (HOG-дескриптор (Histogram of Oriented Gradients, гистограмма ориентированных градиентов)) [3] и его классификации. Данный метод был взят за основу детектора. В качестве классификатора, определяющего принадлежность объекта к группе «Человек» или «Фон», применили метод опорных векторов (SVM).

Для повышения вероятности распознавания присутствия людей в кадре алгоритм определения фигуры человека содержит ряд шагов (этапов):

1. Запускается детектор движения. Поскольку анализу подвергается изображение большого размера (1Мпиксель и более) то выполнить алгоритм распознавания человека методом HOG на всем кадре за приемлемое время практически невозможно. При этом для каждого кадра выпол-

няется сравнение с фоновым изображением и выделяются участки, в которых разница фонового изображения и текущего кадра превышает заданный порог. Затем модель фона обновляется. Участки движения сегментируются, и часть из них отбрасывается в случае, если их размер не удовлетворяет заранее заданному размеру человека.

2. Выделенные прямоугольники подаются на детектор, в котором происходит перемасштабирование исходя из заранее заданных минимального и максимального размеров человека. При этом используются данные о положении линии горизонта на изображении, а также допущение, что размеры человека линейно растут от минимальных (на дальнем плане, вблизи линии горизонта) до максимальных (на ближнем плане).

3. Для каждого окна строится вектор признаков из гистограмм направленных градиентов и производится классификация, которая определяет, находится ли в данном окне фигура человека или нет. Внутри каждой области происходит анализ на наличие человека из серии окон различного масштаба с шагом в 1%. Применение описанного подхода позволяет значительно сократить количество поисковых окон. Кроме того, за счет сужения области поиска людей уменьшается количество ложных срабатываний алгоритма, что приводит к повышению качества работы системы в целом.

4. Все детектируемые объекты записываются в буфер, в котором для каждого объекта хранятся:

- текущие координаты;
- история перемещения (трекинг) с данными о статусе («HOG» – если на данном кадре объект определен с помощью алгоритма HOG, или «трекинг» – если объект не подтвердился алгоритмом HOG и сопровождался при помощи алгоритма оптического потока);
- время последней детекции алгоритмом HOG;
- изображение объекта.

Для каждого последующего кадра происходит перебор всех ранее распознанных объектов и для каждого объекта снова происходит поиск алгоритмом HOG в окне большего размера. Если объект снова распознан, обновляются данные по объекту в буфере, в противном случае запускается процедура трекинга и в буфере обновляются данные по объекту с указанием в статусе, что произведен трекинг.

Также с заданной периодичностью запускается детектор движения и выполняются шаги 1–3, а в буфер добавляются только новые объекты.

При анализе всех объектов, содержащихся в буфере, отображаются только объекты, для которых было несколько подтверждений распознавания алгоритмом HOG и позиции распознаваний находились не в одном месте.

Данные ограничения позволили повысить вероятность распознавания людей на сложном рельефе (например: свалка металлолома) и исключить ложные срабатывания, неизбежно появляющиеся в таких случаях. На рис. 1 приведен один из видеок кадров во время детекции и трекинга людей.



Рис. 1

В детекторе движения применяется алгоритм вычитания фона из текущего кадра. Модель фона строится по следующей формуле:

$$F_k(x, y) = \alpha I_k(x, y) + (1 - \alpha)F_{k-1}(x, y),$$

где α – задает скорость обновления фона; $I_k(x, y)$ – значение интенсивности освещения в текущей точке; $F_{k-1}(x, y)$ – предыдущее значение модели фона.

На каждом шаге происходит сравнение интенсивностей в каждой точке модели фона и текущего кадра и при превышении формируется бинарное изображение. Далее над бинарным изображением проводятся морфологические операции эрозии и дилатации, после чего запускается алгоритм сегментации, который возвращает отдельные области.

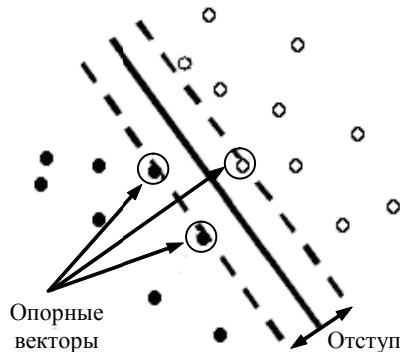


Рис. 2

В качестве алгоритма машинного обучения используется линейная машина опорных векторов (Linear SVM) [4]. Отметим, что натренированной моделью линейного SVM является плоскость, наилучшим образом разделяющая объекты от негативов в обучающей выборке (рис. 2). На рисунке отображены объекты, разделенные гиперплоскостью, которая отделяет группу объектов, имеющих различную классовую принадлежность. В этом примере объекты относятся либо к классу ЧЕРНОГО, либо БЕЛОГО цвета. Разделительная линия определяет границу, с правой стороны от которой все объекты БЕЛЫЕ, а с левой – ЧЕРНЫЕ. Любой новый объект, попадающий справа, классифицируется как БЕЛЫЙ (или ЧЕРНЫЙ, если он попадает слева от разделительной линии). Векторы, лежащие ближе всех к разделяющей гиперплоскости, называются опорными векторами (support vectors).

Для обучения классификатора требуется сформировать обучающую выборку, содержащую признаковые описания изображений с объектом и без него. Так как источником таких изображений, как правило, являются фотографии, большую часть которых занимает фон, а не сам объект, большинство окон детектирования не будут содержать объект. В связи с этим требуется аккуратная балансировка прецедентов различных классов в обучающей выборке. Одним из подходов к обучению классификатора является итерационный

алгоритм добавления признаковых описаний изображений фона бутстраппинг (Bootstrapping) [5]. На нулевой итерации в обучающую выборку добавляются признаковые описания всех изображений с объектом, приведенных к нужному размеру, и выбранные случайным образом области изображений с фоном. С использованием полученной выборки осуществляется обучение классификатора, после чего выполняется детектирование объектов на изображениях фона, и выборка дополняется признаковыми описаниями ложных срабатываний детектора. Классификатор обучается на измененной выборке заново. Данный процесс повторяется несколько раз. Подобная процедура позволяет автоматически отрегулировать количество прецедентов в обучающей выборке, соответствующих изображениям без искомого объекта, и, тем самым, уменьшить количество ложных срабатываний полученного детектора.

Использование алгоритма HOG совместно с алгоритмами трекинга и детектором движения, а также информация о пространственных размерах фигуры человека и ограничения на возможные геометрические перемещения позволили создать детектор распознавания людей на сложном рельефе с вероятностью правильного распознавания не ниже 90 % и минимизировать ложные срабатывания, неизбежно появляющиеся в системах распознавания образов.

СПИСОК ЛИТЕРАТУРЫ

1. Viola P., Jones M. Rapid object detection using a boosted cascade of simple feature // Conference on Computer Vision and Pattern Recognition (CVPR). 2001.
2. Enzweiler M., Dariu M. Gavrilă. Monocular Pedestrian Detection: Survey and Experiments // Pattern Analysis and Machine Intelligence. 2009.
3. Dalal N., Triggs B. Histograms of oriented gradients for human detection // Conference on Computer Vision and Pattern Recognition (CVPR). 2005.
4. Vapnik V. N. The Nature of Statistical Learning Theory. Springer-Verlag, 1995.
5. Freund Y., Schapire R. E. A Short Introduction to Boosting // J. of Japanese Society for Artificial Intelligence. 1999. № 14(5). P. 771–780.

I. A. Goroshkov, S. A. Kulakov

DETECTOR OF PRESENCE OF PEOPLE IN THE DETECTION BLOCK

The article presents an algorithm for recognizing the presence of a person in the frame. The advantages of the selected algorithm for determining people. Mentions the possibility of applying this algorithm to ensure the safety and security of goods and materials to suit all types of businesses and activities.

An algorithm for recognizing people, HOG, machine learning, tracking

УДК 519.218.82

Н. И. Орешко

Восстановление закона изменения гетероскедастического шума при траекторных измерениях на основе вейвлет-технологии

Рассматривается проблема очистки от шума траекторных и телеметрических данных при наличии гетероскедастического шума. В отличие от более ранних статей [1]–[4], в которых был рассмотрен специальный класс модели изменения шума от времени, а именно кусочно-постоянная модель, здесь рассматривается общий случай изменения дисперсии шума во времени по неизвестному закону при естественных ограничениях на гладкость такой модели. Задача решается восстановлением закона изменения среднеквадратической погрешности нестационарного шума.

Очистка от шума, гетероскедастический шум, вейвлет-преобразование

Постановка задачи. Предположим, имеется набор некоторых измеренных данных y_i ($i=1, \dots, n$), выбранных на равномерной временной сетке t_i ($i=1, \dots, n$). Рассматривается аддитивная модель вида

$$y_i = f(t_i) + \sigma_i \xi_i \quad (i=1, \dots, n),$$

где $f(t_i)$ – истинные значения наблюдаемого процесса в моменты времени t_i ($i=1, \dots, n$), а случайные погрешности ξ_i независимы и одинаково распределены по нормальному закону $\xi_i \sim N(0,1)$, но дисперсия $\text{var}(\varepsilon_i) = \text{var}(\sigma_i \xi_i) = \sigma_i^2$, соответствующая случайной погрешности $\varepsilon_i = \sigma_i \xi_i$, изменяется во времени. Предполагается также, что закон изменения $\sigma_i = \sigma(t_i)$ от времени описывается, возможно, сложной, но гладкой функцией времени. Необходимо оценить $f(t_i)$ по результатам измерений y_i .

В такой постановке в статистике используется специализированный термин, а именно гетероскедастичность, означающий неоднородность наблюдений и выражающийся в неодинаковой (непостоянной) дисперсии случайной ошибки регрессионной модели. Гетероскедастичность противоположна *гомоскедастичности*, означающей однородность наблюдений, т. е. постоянство дисперсии случайных ошибок модели.

При рассмотрении классической регрессионной модели метод наименьших квадратов (МНК) дает наилучшие линейные несмещенные оценки лишь при выполнении ряда предпосылок, одной из которых является постоянство дисперсии отклонений (гомоскедастичность) для всех наблюдений. При невыполнении данной предпосылки (гетероскедастичности) последствия применения МНК будут следующими:

1. Оценки остаются несмещенными и линейными.
2. Оценки не будут эффективными.
3. Дисперсии оценок будут рассчитываться со смещением.
4. Статистические выводы, полученные при стандартных проверках качества оценок, могут быть ошибочными.

Для проверки наличия гетероскедастичности применяют, например, статистические тесты Уайта, Голдфелда–Куандта, Бройша–Пагана, Парка, Глейзера, Спирмена. В данной статье использован критерий Спирмена [5]. Для проверки гипотезы на основе данного критерия вначале производится выделение случайной погрешности ε_i ($i=1, \dots, n$) с помощью специализированного вида избыточного вейвлет-преобразования, а именно максимально накладываемого дискретного вейвлет-преобразования (МНДВП). Выделенная случайная погрешность на всем интервале наблюдения и подвергается проверке с использованием данного критерия.

Тест ранговой корреляции Спирмена – непараметрический статистический тест, позволяющий проверить гетероскедастичность случайных ошибок. Особенность теста заключается в том, что не конкретизируется форма возможной зависимости дисперсии случайных ошибок модели от той или иной переменной.

Для проверки гипотезы гетероскедастичности выделенного случайного процесса погрешностей определяется коэффициент ранговой корреляции Спирмена:

$$\hat{\rho} = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)},$$

где d_i – разность рангов переменных ε_i и n (n соответствует рангу временной последовательности). При справедливости нулевой гипотезы (отсутствии гетероскедастичности, т.е. в данном случае – равенство нулю истинного значения коэффициента ранговой корреляции Спирмена ρ) статистика $\hat{\rho}\sqrt{n-1}$ асимптотически (т.е. при достаточно большом n) имеет стандартное нормальное распределение $N(0,1)$. Соответственно, если значение указанной статистики больше критического значения этого распределения (при данном уровне значимости), то гетероскедастичность признается значимой. В противном случае гетероскедастичность незначима.

Если коэффициент корреляции $\hat{\rho}$ для генеральной совокупности равен нулю, то статистика

$$t = \frac{\hat{\rho}\sqrt{n-2}}{\sqrt{1-\hat{\rho}^2}}$$

имеет распределение Стьюдента с числом степеней свободы $\nu = n - 2$. Следовательно, если наблюдаемое значение t -статистики превышает $t_{кр} = t_{\frac{\alpha}{2}, n-2}$ (определяемое по таблице критиче-

ских точек распределения Стьюдента), то необходимо отклонить гипотезу о равенстве нулю коэффициента корреляции $\hat{\rho}$, а следовательно, и об отсутствии гетероскедастичности. В противном случае гипотеза об отсутствии гетероскедастичности принимается.

Обзор существующих подходов. Стандартным подходом при работе с гетероскедастичным шумом является предположение о виде его зависимости от времени (например, в виде некоторой простой параметрической зависимости –

чаще всего линейной) с последующей оценкой параметров предложенной модели. Если же модель изменения неизвестна, то используют методы вычисления дисперсии погрешностей на скользящем интервале. Для оценки σ могут применяться также различные методы сглаживания (ядерные методы, сглаживание сплайнами). Таким образом, в данном случае очистка от гетероскедастического шума основана на применении различной ширины полосы в непараметрической ядерной регрессии, что приводит к большему сглаживанию участков сигнала с высокой дисперсией. В [6] дисперсионная функция оценивается с использованием локального полиномиального сглаживания квадратов остатков. Другой подход, приведенный в [7], оценивает $\log(\sigma^2(t))$, используя сглаживание любым непараметрическим регрессионным методом логарифма квадрата остатков $\log(y(i) - \hat{f}(i))^2$. Далее дисперсия определяется из равенства $\hat{\sigma}_t^2(t) = e^{\hat{q}(t)}$, где $\hat{q}(t)$ – сглаживающая оценка $\log(\sigma^2(t))$. Недостаток этого подхода заключается в том, что логарифм очень малых остатков приводит к большим выбросам. Адаптивное оценивание дисперсионной функции с использованием вейвлет-пороговой обработки описано в [8], где процедура вейвлет-пороговой обработки применяется к квадрату разностей исходных наблюдений. В этом случае задача оценивания дисперсионной функции сводится к оцениванию среднего.

Описанные в литературе методы в основном используют квадраты остатков для оценивания закона изменения шумовой составляющей. Однако они довольно чувствительны даже к небольшим выбросам и точность восстановления модели невысокая.

Алгоритм удаления нестационарного шума, дисперсия которого изменяется по неизвестному гладкому закону. Предлагаемый алгоритм восстанавливает закон изменения среднеквадратичного отклонения шумовой составляющей, используя абсолютные значения вейвлет-коэффициентов, и осуществляет пороговую обработку с учетом восстановленного закона.

Алгоритм включает в себя следующие этапы:

1. Выделение шумовой составляющей с использованием МНДВП.

2. Вычисление модуля выделенной шумовой составляющей во временной области.

3. Определение тренда модуля шума.

4. Пороговая обработка вейвлет-коэффициентов с учетом определенного закона для шумовой составляющей.

5. Обратное вейвлет-преобразование.

Выделение шумовой составляющей. Шумовая составляющая сигнала определяется за счет разделяющей способности вейвлет-преобразования. Шумовые вейвлет-коэффициенты сосредоточены на самых тонких (высокочастотных) уровнях вейвлет-разложения. Мультиразрешающий анализ на основе МНДВП позволяет выделить высокочастотные составляющие во временной области. Эта процедура подобна использованию фильтра с нулевой фазой, поэтому закон изменения СКО шума не искажается за счет несимметричной формы используемого при разложении вейвлет-фильтра. Таким образом, выделенную шумовую составляющую можно представить в виде

$$\tilde{\sigma} = \underbrace{B_1^T W_1}_{D_1} + \underbrace{A_1^T B_2^T W_2}_{D_2} + \dots + \underbrace{A_1^T \dots A_{J-1}^T B_J^T W_J}_{D_J},$$

где $W_1, W_2, \dots, W_J$ – вейвлет-коэффициенты первых J уровней разложения; $D_1, D_2, \dots, D_J$ – проекции на детализирующее пространство соответствующих частотных полос; A, B – матрицы преобразований.

Выделенная составляющая содержит в основном шум, если истинный сигнал является гладким. Для сигналов с особенностями (резкими изменениями, пиками) составляющая содержит кроме шума небольшое число значительно отличающихся по амплитуде вейвлет-коэффициентов, соответствующих этим особенностям. В этом случае для меньшего искажения особенностей при пороговой обработке рекомендуется использовать вейвлет с малой длиной фильтра, т. е. с меньшим носителем.

Определение закона изменения СКО шума.

Шумовая составляющая может изменяться по какому-либо неизвестному закону, однако предполагается, что этот закон описывается некоторой гладкой функцией. Более точно предполагается, что существует по крайней мере вторая непрерывная производная для этой функции.

Закон изменения СКО шума определяется по абсолютному значению выделенной на предыдущем этапе шумовой составляющей с использованием разных подходов, например ядерного сглаживания, разложения по эмпирическим модам, вейвлет-разложения. В данном исследовании высокую эффективность и гибкость показал метод робастной сплайн-аппроксимации [9].

Для окончательного получения оценки закона изменения шума $\hat{\sigma}(t)$ вычисленный тренд нормируется на величину $\sqrt{2/\pi}$, поскольку для среднего значения распределения модуля случайной величины, распределенной по нормальному закону, справедливо соотношение

$$E(\tilde{\xi}) = \sigma\sqrt{2/\pi} \text{ при } \mu = 0,$$

где $\xi \in N(\mu, \sigma)$ и $\tilde{\xi} = |\xi|$ [5].

Восстановленный закон изменения СКО шума может использоваться для очистки от шума исходного сигнала с использованием различных методов. Если применять вейвлет-подход, то вейвлет-преобразование должно применяться и для исходного сигнала, и для восстановленного закона изменения СКО шума. В этом случае порог рассчитывается с учетом вейвлет-преобразования закона СКО шумовой составляющей.

Результаты. Рассмотрим результаты работы предложенного подхода для двух экспериментов. Для первого эксперимента (рис. 1–4) в качестве основной задачи исследовалась возможность восстановления сложного поведения дисперсии шумовой составляющей. Закон изменения шумовой компоненты моделировался в виде синусоиды (рис. 3, а), а на рис. 3, б приведена сама реализация шумового процесса. Сам же сигнал (светлым цветом) и его очистка от шума (темная жирная линия) на основе стандартного избыточного вейвлет-преобразования изображены на рис. 1. Сигнал (светлым цветом) и его очистка от шума (темная жирная линия) на основе предложенной технологии изображены на рис. 2. На рис. 4 приведены этапы работы предложенного алгоритма. На рис. 4, в приведена нормированная восстановленная реализация, на рис. 4, б – аппроксимация (светлым цветом) абсолютного значения выделенного шума (темным цветом), а на рис. 4, а – аппроксимация (светлым цветом) выделенной компоненты (темным цветом).

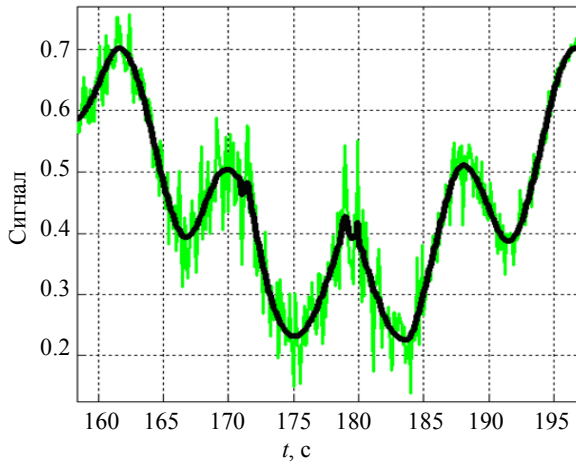


Рис. 1

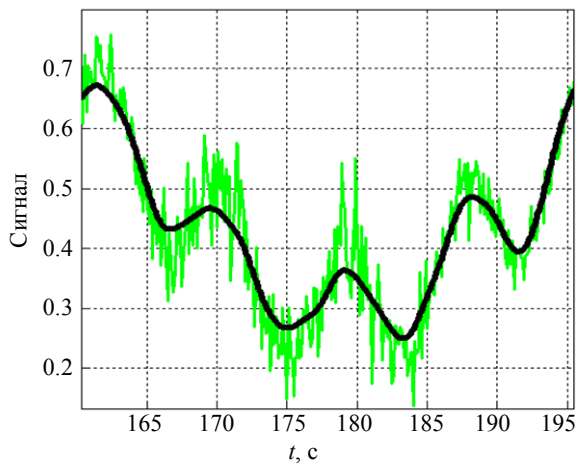
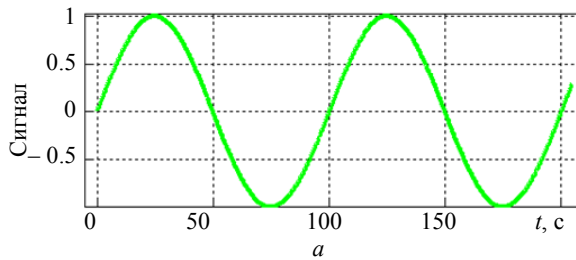


Рис. 2



а

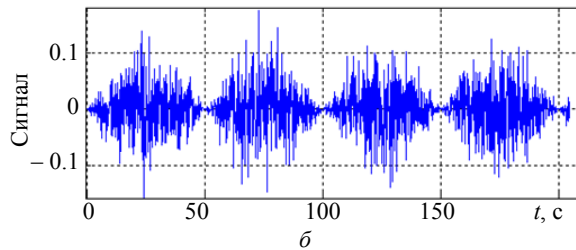
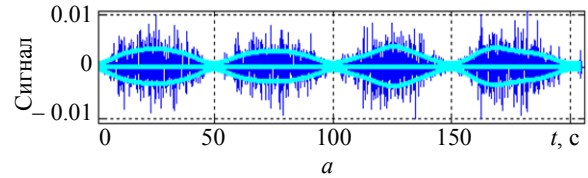


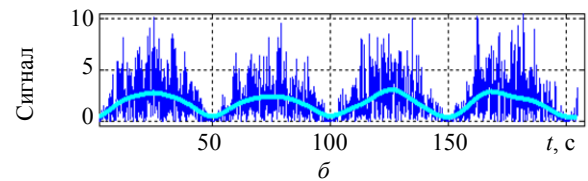
Рис. 3

Основная цель второго эксперимента (рис. 5–10) – показать эффективность работы предложенного алгоритма в задаче очистки сложного сигнала при наличии гетероскедастического шума. Сигнал моделировался в виде волнообразного процесса (рис. 5), а шум – в виде процесса с посто-

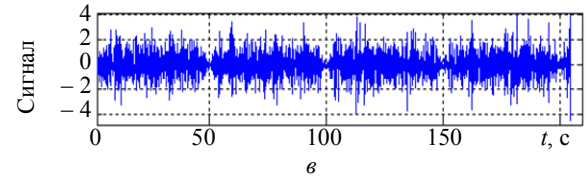
янной единичной дисперсией, существенно возрастающей на конечном интервале наблюдения (рис. 9). На рис. 6 показаны исходный (светлая линия) и очищенный (темная линия) с использованием стандартной вейвлет-технологии сигналы, а на рис. 7 – исходный и очищенный с использованием предложенной вейвлет-технологии сигналы.



а



б



в

Рис. 4

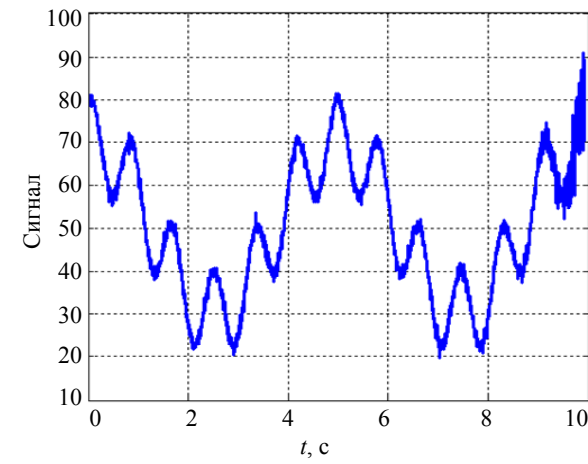


Рис. 5

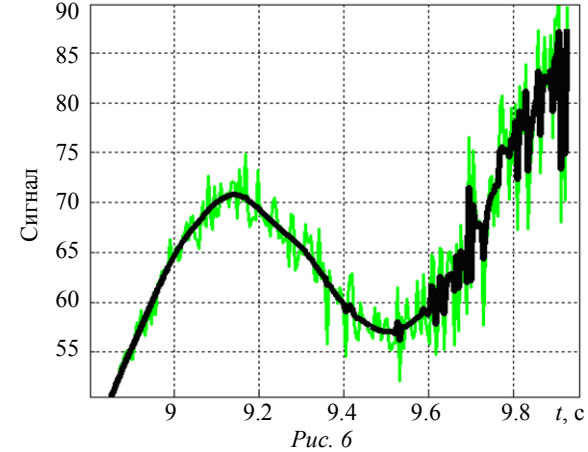


Рис. 6

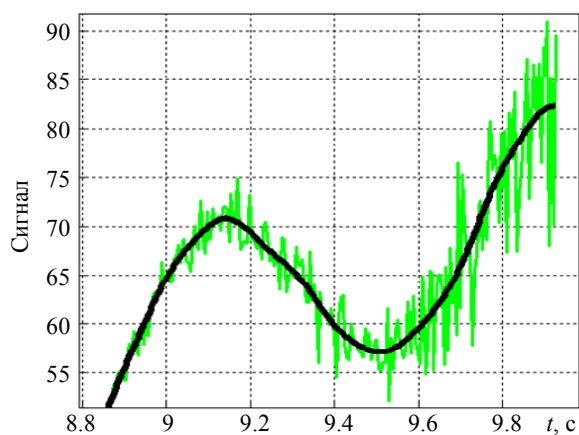


Рис. 7

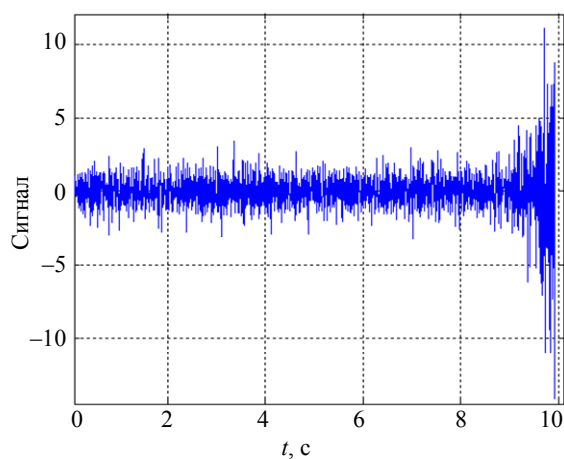
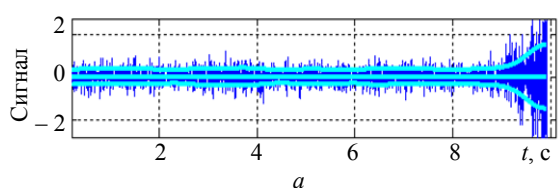
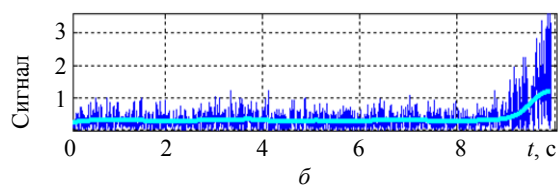


Рис. 9



а



б

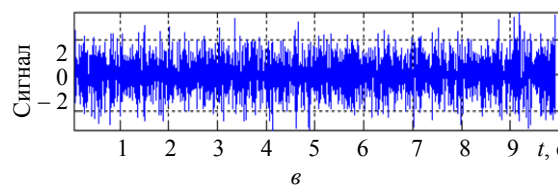


Рис. 8

На рис. 8 приведены этапы работы предложенного алгоритма. На рис. 8, в приведена нормированная восстановленная реализация, на рис. 8, б – аппроксимация (светлым цветом) абсолютного значения выделенного шума (темным цветом), а на рис. 8, а – аппроксимация (светлым цветом) выделенной компоненты (темным цветом).

На рис. 9 приведен выделенный алгоритмом шумовой процесс.

Из сравнения рис. 6 и 7, где приведены в увеличенном масштабе конечные части исходного и очищенного от шума сигналов, видно радикальное улучшение качества очистки от шума на основе предложенной технологии.

Таким образом, алгоритм проявляет высокую эффективность при наличии гетероскедастического шума с неизвестным законом изменения для сигналов со сложным поведением.

СПИСОК ЛИТЕРАТУРЫ

1. Князева Т. Н., Новиков Л. В., Орешко Н. И. Удаление нестационарного шума из экспериментальных данных // Науч. приборостроение. 2008. Т. 18, № 2. С. 61–65.
2. Knyazeva T. N., Oreshko N. I. Wavelet Denoising of Experimental Data with Non-stationary Noise // Тр. Междунар. конф. "Wavelets and Applications". 2009. Р. 28–30.
3. Орешко Н. И., Князева Т. Н. Вейвлет-технология анализа и очистки от шума сигналов // Цифровая обработка сигналов. 2008. № 3. С. 21–25.
4. Орешко Н. И., Князева Т. Н. Очистка от шума траекторных данных при наличии гетероскедастических погрешностей // Изв. СПбГЭТУ "ЛЭТИ". 2007. Вып. 2. С. 31–40.
5. Кобзарь А. И. Прикладная математическая статистика. М.: ФИЗМАТЛИТ, 2006.
6. Fan J., Yao Q. Efficient estimation of conditional variance functions in stochastic regression // Biometrika. 1998. № 85. Р. 645–660.
7. Yu K., Jones M. Likelihood-based local linear estimation of the conditional variance function // J. of the American Statistical association. 2004. № 99. Р. 528–555.
8. Cai T., Wang L. Adaptive variance function estimation in heteroscedastic nonparametric regression // Ann. Statist. Vol. 36, № 5. Р. 2025–2054.
9. Garcia D. Robust smoothing of gridded data in one and higher dimensions with missing values // Computational Statistics & Data Analysis. 2010.

N. I. Oreshko

ESTIMATING LAW OF VARIATION OF HETEROSCEDASTIC NOISE IN TRAJECTORY MEASUREMENTS ON THE BASIS OF WAVELETS

In this paper we describe a two-step method of denoising of complex data in heteroscedastic nonparametric regression. At the first step we extract using wavelets a noise component and estimate for its absolute values using robust spline fitting the model for the variance function. At the second step we employ the estimated variance function for wavelet coefficients thresholding on the basis of overlap discrete wavelet transform.

Denoising, heteroscedastic noise, wavelet transform

УДК 621.397.01

С. С. Голяк, С. А. Гордеев, И. А. Горошков, Г. В. Разумовский

Комплексная система видеонаблюдения за технологическими процессами

Описывается архитектура комплексной системы видеонаблюдения, рассматриваются вопросы ее взаимодействия с различными информационными системами промышленного предприятия и использования для контроля технологических процессов.

Система видеонаблюдения, детектор видеоаналитики, сервер обработки событий, контроллер видеосигналов

Необходимость слежения и контроля за технологическими процессами обуславливается многими факторами, основные из которых – безопасность производства и предотвращение хищений товарно-материальных ценностей. Технологический процесс можно рассматривать как последовательность некоторых технологических событий, каждое из которых может контролироваться отдельными средствами и методами. Полностью контролировать технологический процесс одним средством практически невозможно, поэтому для контроля технологических событий используются различные датчики, сканеры, магнитные метки и считыватели, а также целые автоматизированные системы управления. Относительно недавно к этому списку добавились и видеокамеры, которые в совокупности объединены в систему видеонаблюдения.

Специализированное программное обеспечение способно анализировать получаемую с видеокамеры картинку и при необходимости генерировать соответствующие события. Видеокамеры выгодно отличаются от других средств контроля тем, что их можно установить почти в любом месте, даже там, где нельзя поставить датчики, либо

датчики в этом месте будут плохо работать. Также часто предпочтение отдается видеокамерам ввиду их относительной дешевизны по сравнению с дорогостоящими датчиками, метками и сканерами.

Система видеонаблюдения вместе со специализированным программным обеспечением анализа изображения составляют систему видеоаналитики. Система видеоаналитики предоставляет следующие возможности:

- идентификация объекта, т. е. принятие решения о том, что в кадре присутствует искомый объект;
- определение направления движения объекта;
- распознавание пересечения объектом заданной линии;
- распознавание текста, в частности, номеров автомобилей и железнодорожных составов.

Таким образом, система видеоаналитики совместно с другими датчиками и средствами контроля располагают исходными данными для отслеживания технологического процесса комплексной системой видеонаблюдения. Слежение происходит сопоставлением полученных событий от системы видеоаналитики и других средств контроля с некоторыми правилами, заданными в системе заранее.

Комплексная система видеонаблюдения промышленного предприятия – это распределенная система, состоящая в общем случае из следующих компонентов (рис. 1): сервер обработки событий, база данных, сервис-деск, интерактивная карта, источники событий и сервисная шина предприятия. Источники событий могут включать в себя автомобильные весы, систему контроля доступа (СКД), ERP-систему подготовки производства, бюро пропусков и систему оперативного слежения за передвижением автомобилей по географическим GPS-координатам.

Сервер обработки событий является главным компонентом системы. Его задача заключается в получении сообщений о событиях от источников событий, выстраивании полученных сообщений в хронологическом порядке и проверке получившейся цепочки сообщений согласно прописанным заранее правилам, соответствующим заданным технологическим процессам. При отклонении от правил формируется так называемый инцидент, который отправляется в систему учета инцидентов – сервис-деск. В этом случае в пользовательском интерфейсе сервис-деска создается новая запись, на которую обращает внимание диспетчер и принимает соответствующее решение.

Связующим компонентом системы является сервисная шина предприятия (ESB, enterprise service bus), которая позволяет обмениваться сообщениями между компонентами распределенной системы по различным протоколам, включая JMS, HTTP, SMTP, SOAP и др. В рассматриваемой системе сервисная шина предприятия реализована на основе открытого программного обеспечения Apache ServiceMix и Apache ActiveMQ.

База данных представляет собой хранилище истории зафиксированных событий для последующего их анализа в ситуации, когда решение по

рассматриваемому сервером событию зависит от предыстории зафиксированных ранее событий. Также в базе данных хранится справочная информация об объектах наблюдения, наименовании видеокамер и формируемых инцидентах, нормативах, регламентирующих технологические процессы предприятия.

Для просмотра «живого» видео с камер наблюдения используется интерфейс интерактивной карты, в котором пользователь по схематическому плану предприятия может найти и выбрать необходимую для просмотра камеру. Также в интерактивной карте предусмотрен режим видеостены, при котором на экране в табличном виде отображаются видео сразу с нескольких камер одновременно, что делает возможным просмотр видео издалека сразу несколькими диспетчерами.

Все камеры, установленные на предприятии, подключаются к видеоархивам, а часть камер, которые используются для видеоаналитики, подключаются к контроллерам видеосигналов (КВС). На КВС установлено специализированное программное обеспечение (детекторы), анализирующее изображение с видеокамер. В среднем к одному КВС можно подключить до 12 камер, точное количество подключаемых камер зависит от решаемых детектором задач и может быть от 4 до 20.

КВС представляет собой аппаратно-программный комплекс, установленный на компьютере с многоядерным процессором для организации параллельной работы детекторов видеоаналитики (рис. 2). Детекторы, работающие на одном КВС, независимы между собой, и каждый детектор обслуживают свою камеру, поэтому их можно запускать как отдельные процессы. Управляет запуском детекторов специальный модуль, который

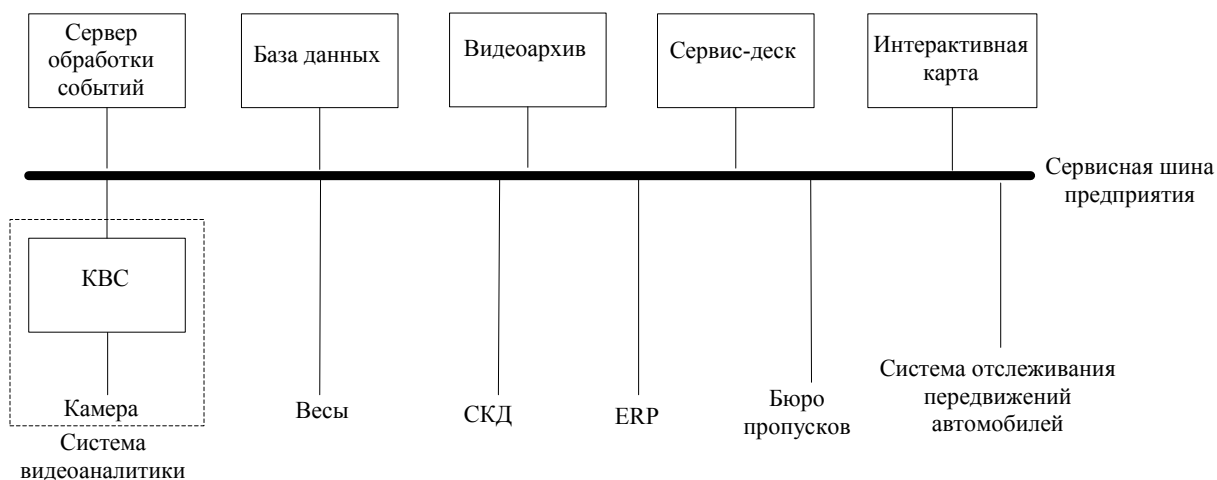


Рис. 1

установлен в виде службы и запускается автоматически при загрузке операционной системы. При старте КВС модуль обращается к файлу конфигурации и определяет, с какими параметрами необходимо запускать детекторы и к каким камерам они должны подключаться. При выявлении события детектор обращается к модулю отправки сообщений для передачи соответствующего сообщения на сервер событий. Помимо этого, КВС с определенным интервалом времени отправляет сообщения типа PING, позволяющие серверу обработки событий определять работоспособность КВС. Для отсчета интервала используется таймер.

На КВС могут быть установлены следующие детекторы видеоаналитики:

- детектор определения противоправных действий по отношению к камерам (закрытие объектива, засветка объектива, сдвиг камеры);
- детектор определения направления движения;
- детектор определения проезда автомобиля с распознаванием его государственного номера;
- детектор определения проезда железнодорожного состава с подсчетом числа вагонов и определением их номеров;
- детектор распознавания присутствия людей.

Последовательность поступления сообщений от различных источников можно проследить на примере технологического процесса перевозки груза со склада в цех (рис. 3).

Этот технологический процесс контролируется на основе данных, поступающих от видеокамеры, СКД, бюро пропусков, весов, ERP-системы

и системы слежения за автомобилями. Эти данные могут поступать в реальном времени либо с запаздыванием. Если данные запаздывают, то обработка сообщения откладывается. Контроль данного технологического процесса выполняется в следующей очередности. При въезде на склад сравнивается номер автомобиля, распознанный видеокамерой, с номером, полученным от СКД, и проверяется правомерность проезда автомобиля обращением к системе «Бюро пропусков». После заезда на склад автомобиль взвешивается на весах, чтобы узнать массу незагруженного автомобиля. Далее его загружают материалом и снова взвешивают. Каждое взвешивание автомобиля фиксируется, и сведения о массе автомобиля отправляются серверу событий. Помимо сервера событий информация о массе отправляется в ERP-систему подготовки производства, в которой формируется накладная для указанного автомобиля. В накладной указывается номер автомобиля и масса перевозимого груза, высчитываемая как разница масс загруженной и незагруженной машины. После провешиваний автомобиль выезжает со склада и отправляется в цех. По GPS-координатам, полученным от системы отслеживания передвижений автомобилей, можно выявить отклонение машины от маршрута. Основными задачами контроля данного технологического процесса являются фиксация времени нахождения автомобиля в пути и контроль массы материала, привезенного в цех. Эти парамет-

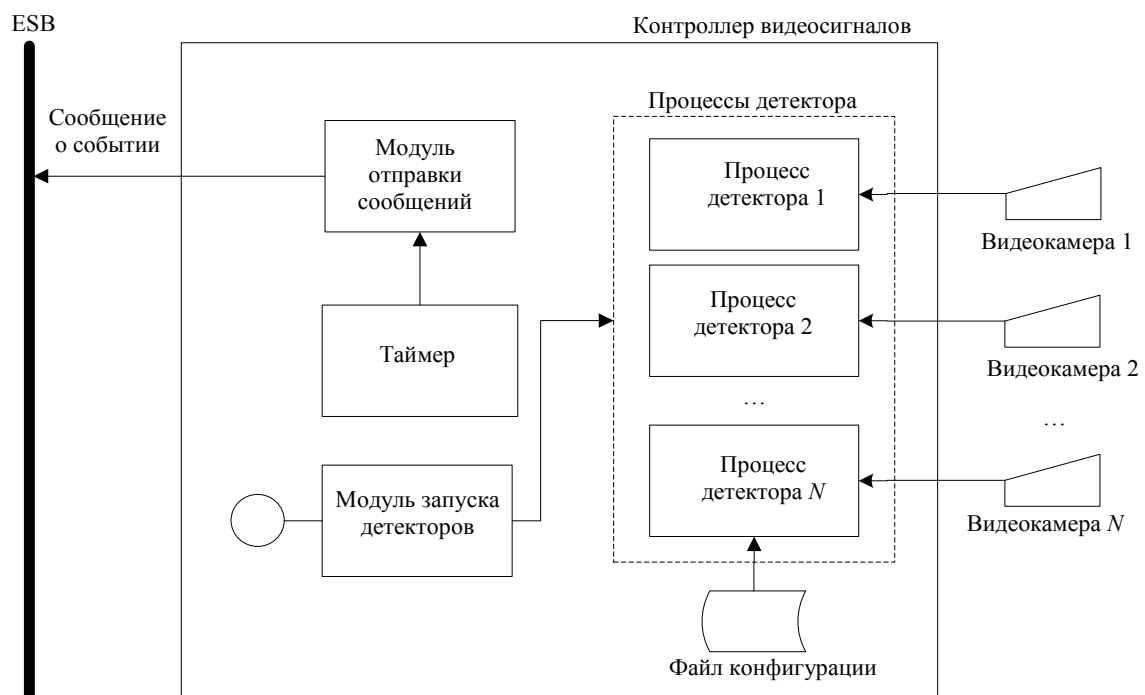


Рис. 2

ры проверяются по прибытию автомобиля в производственный цех, в котором он проходит те же операции, что и на складе, только в обратном порядке.

Все события, фиксируемые системой видеонаблюдения и другими источниками событий, отправляются серверу обработки событий для дальнейшего анализа и принятия решения о генерации инцидента. Процесс обработки сообщений на сервере событий представлен на рис. 4.

При получении очередного сообщения из сервисной шины сервер определяет тип получаемого сообщения и сохраняет содержимое сообщения в таблице базы данных. В зависимости от типа сообщения оно может быть передано обработчику, в котором заложена логика обработки данного типа сообщения, либо сообщение может инициировать создание сервером отложенного события, которое будет им обработано спустя заданное время. В первом случае обработчик при необходимости может обратиться к таблице с полученными сообщениями для получения предыстории, которая влияет на принятие решения о генерации инцидента (например, для контроля массы привезенного в цех груза необходимо знать предыдущее значение массы на складе и при отклонении массы на заданное значение генерируется инцидент). Второй вариант характерен для проверки различных временных интервалов (например, времени нахождения автомобиля в пути). В этом случае задействуется модуль «Таймер», который по истечении заданного интервала времени сгенерирует отложенное сообщение в сервисную шину, и сервер снова просчитает его.



Рис. 3

В случае принятия сервером решения о генерации инцидента с помощью сервисной шины вызывается веб-сервис, создающий новый инци-

ESB

Сервер обработки событий

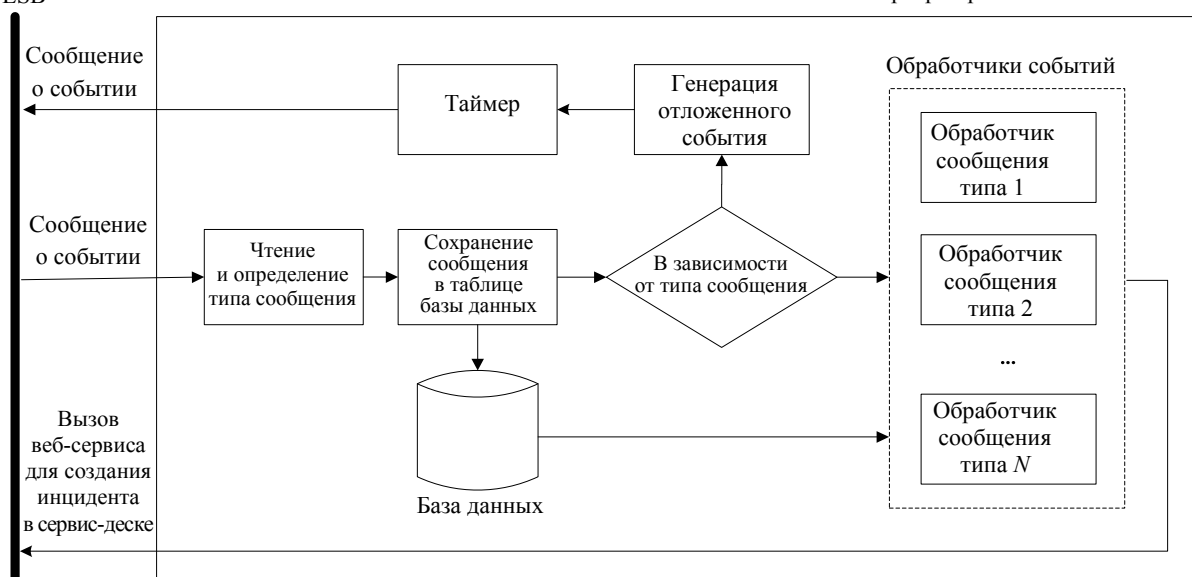


Рис. 4

дент в сервис-деске. К инциденту прикрепляются соответствующие атрибуты (например, номер автомобиля, масса автомобиля, время в пути и т. д.), а также (при возможности) подтверждающие инцидент кадры с видеокамер. Помимо этого диспетчер может просмотреть видеоролик зафиксированного события, который берется из архива системы видеонаблюдения. Таким образом, диспетчер, заметив новый инцидент, может сразу понять причину его создания и принять соответствующее решение.

В заключение следует отметить, что интеграция системы видеоаналитики с датчиками и информационными системами предприятия в единую комплексную систему видеонаблюдения позволяет контролировать всю цепочку технологического процесса в реальном масштабе времени и быстро принимать решения в зависимости от типа и важности зафиксированных событий и инцидентов.

S. S. Golyak, S. A. Gordeev, I. A. Goroshkov, G. V. Razumovsky

INTEGRATED VIDEO SURVEILLANCE SYSTEM FOR TECHNOLOGICAL PROCESS MONITORING

This paper is focused on a monitoring of technological processes via integrated video surveillance system. Authors consider software architecture issues, problems of interaction with other enterprise information systems and ways of usage video surveillance system for technological process monitoring.

Video Surveillance System (CCTV), Video Content Analytics (VCA), Event Processing Server, Video Controller

УДК 512.2

Н. А. Жукова, И. С. Соколов, А. В. Экало

Метод нечеткого сравнения телеметрических параметров на основе алгоритмов интеллектуального анализа данных

Предлагается метод нечеткого сравнения медленно меняющихся телеметрических параметров на основании символического представления с использованием взвешенного редакционного расстояния. Применение редакционного расстояния позволяет сравнивать параметры, смещенные друг относительно друга как по оси времени, так и по оси значений, а также не учитывать несущественные отклонения в поведении параметров, вызванные влиянием внешних факторов.

Интеллектуальный анализ данных, символическое представление, телеметрические параметры

Содержание современной телеметрии составляет широкий круг проблем, связанных с получением, преобразованием, передачей и обработкой телеметрической информации, используемой при управлении удаленными объектами, определении их состояния или при изучении физических процессов в местах, где непосредственное присутствие наблюдателя затруднено или невозможно. Под телеметрируемым параметром (ТПП) понимается показатель физического процесса, события или явления, значение или поведение которого подлежит измерению или контролю телеметрической системой. В зависимости от скорости

изменения во времени ТМП делятся на медленно меняющиеся (ММТП) и быстро меняющиеся (БМТП). Первые характеризуются шириной спектра от 0 до 50 Гц, а вторые имеют верхнюю границу спектра от единиц до десятков и сотен килогерц [1]. В статье рассматриваются только параметры, принадлежащие к классу ММТП.

Одной из основных задач в обработке телеметрии является задача выявления отклонений в поведении ТМП. Наиболее часто используемый подход, основанный на допусковом контроле значений, является низкоэффективным [1] в связи с тем, что параметры содержат большое количество

шумов и выбросов, значения которых выходят за границы допускового интервала. Для объектов, которые в рамках разных опытно-испытательных работ (ОИР) имеют схожее поведение, в качестве альтернативного может быть использован подход, основанный на сравнении параметров с набором эталонных параметров. Решение задачи сравнения временных рядов предполагает решение двух подзадач: определение способа формализованного описания временных рядов и метода их сравнения. Широко применяемые в настоящее время высокоуровневые представления временных рядов [2]–[8] и методы работы с ними не обеспечивают решение задачи сравнения телеметрических параметров. Основная проблема, возникающая при обработке телеметрических параметров, заключается в том, что параметры являются сложными нестационарными временными рядами, содержащими резкие переходы, соответствующие изменению состояния телеметрируемого объекта. Кроме того, в рамках разных ОИР время нахождения объекта в одном и том же состоянии может варьироваться, а значения одного и того же параметра могут передаваться с различным коэффициентом усиления исходного сигнала.

В статье предлагается метод сравнения ММТП, основанный на использовании символьного представления временных рядов, применение которого позволяет эффективно решить проблемы, возникающие при обработке параметров.

Постановка задачи. Телеметрические параметры представляют собой временной ряд вида

$$C = c_1, \dots, c_n,$$

где c_i – измерение параметра в момент времени t_i ; n_i – количество измерений параметра. С информационной точки зрения наибольшую ценность для ММТП представляют амплитудные составляющие параметра, а именно абсолютная величина, скорость, направление и форма изменения параметра.

ММТП обладают кусочно-постоянной моделью поведения [1], которая предполагает описание временного ряда в виде последовательности участков, каждый из которых обладает набором постоянных характеристик:

$$M = \langle M_1(\alpha_1, \beta_1, \dots, \omega_1), \dots, M_k(\alpha_k, \beta_k, \dots, \omega_k) \rangle,$$

где $M_i(\alpha_i, \beta_i, \dots, \omega_i)$ – описание поведения параметра на i -м участке; $\alpha_i, \beta_i, \dots, \omega_i$ – набор характеристик, описывающих i -й участок. Формируемое описание временного ряда должно удовлетво-

рять условию: $\sum_{i=1}^k |M_i| = n$, где $|M_i|$ – длина i -го участка.

Для выявления отклонений в поведении ТМП оно сравнивается с эталонным поведением Σ , определенным для каждого параметра. Эталонное поведение задается с помощью одного или нескольких временных рядов:

$$\Sigma = S_1, \dots, S_m,$$

где $S_i = s_1^i, \dots, s_n^i$ – временной ряд, определяющий один из возможных вариантов поведения (шаблонов поведения) параметра, соответствующего штатному состоянию телеметрируемого объекта; m – общее количество шаблонов.

Необходимо разработать метод сравнения ММТП, который обеспечит:

а) возможность формирования высокоуровневого представления параметров $\hat{C}$ на основе исходного временного ряда:

$$\Phi: C \rightarrow \hat{C}, \quad \hat{C} = \hat{c}_1, \dots, \hat{c}_w, \quad n \gg w,$$

где Φ – оператор, выполняющий перевод C в $\hat{C}$; $\hat{c}_i$ – i -й элемент высокоуровневого представления; w – общее количество элементов высокоуровневого представления;

б) возможность расчета расстояния d между парами параметров, для которых сформированы высокоуровневые представления: $\hat{C}: d(\hat{C}_i, \hat{C}_j)$,

где $\hat{C}_i$ и $\hat{C}_j$ – описания i -го и j -го параметров соответственно.

Описание метода символьного представления ТМП. В основу метода сравнения ММТП положен метод SAX (Symbolic Aggregate approXimation), предложенный Е. Кеогом [9]–[11]. Метод SAX предполагает формирование символьного представления исходного временного ряда через промежуточное кусочно-постоянное представление (Piecewise Aggregate Approximation, PAA). Для построения PAA-представления данные разбиваются на W участков (сегментов) одинаковой длины. При формировании описания временного ряда рассчитывается среднее значение для каждого из сегментов. Применение PAA-представления в алгоритме SAX позволяет сократить размерность обрабатываемых данных и обеспечивает наличие нижней границы вычисляемых значений расстояний между временными рядами [12], [13].

Алгоритм 1. Построение РАА-представления.

Вход: $C = c_1, \dots, c_n$ – исходный временной ряд, w – количество. Выход: $\bar{C} = \bar{c}_1, \dots, \bar{c}_n$.

1 $l \leftarrow \text{floor}\left(\frac{n}{w}\right)$

2 для $i = 1$ до w

3 $\bar{c} = \frac{1}{l} \left(\sum_{j=l(i-1)+1}^{l \cdot i} c_j \right)$

4 конец для

При построении символьного представления на основе РАА-представления делается предположение о том, что нормированные значения исходного временного ряда имеют распределение, близкое к нормальному. Для сопоставления каждому из значений РАА-представления символа диапазон значений разбивается на равные области (уровни значений): $B = \beta_1, \dots, \beta_{\alpha-1}$, где B – множество уровней разбиения; β_i – граница между $(i-1)$ -м и i -м уровнями; $[\beta_{i-1}, \dots, \beta_i]$ – i -й уровень разбиения. Все уровни разбиения имеют одинаковую площадь, равную $1/\alpha$. Кроме того, предполагается, что $\beta_0 = -\infty$, $\beta_\alpha = +\infty$. Каждому уровню разбиения сопоставляется уникальный символ из конечного алфавита. Построенное разбиение обеспечивает возможность формирования однозначного символьного описания временного ряда.

При построении символьного представления телеметрических параметров с использованием алгоритма SAX необходимо учесть следующие особенности:

а) распределение исходных значений ММТП не подчиняется нормальному закону;

б) символьные описания временных рядов при использовании различных коэффициентов усиления будут существенно различаться;

в) разбиение на равные интервалы при формировании промежуточного описания ММТП может привести к тому, что в один сегмент будут включены измерения, соответствующие различным состояниям объекта и переходу между состояниями.

Учет перечисленных особенностей работы с телеметрическими параметрами обеспечивается за счет внесения следующих модификаций в алгоритм SAX:

а) осуществляется предварительная нормализация параметров таким образом, чтобы значения нормализованного временного ряда находились в интервале $[-1, 1]$; при нормализации временного

ряда в качестве минимального и максимального значений рассматривается медиана, вычисленная среди K максимальных и минимальных значений соответственно, что позволяет существенно снизить влияние шумов и выбросов на работу алгоритма;

б) в качестве граничных значений сегментов рассматриваются моменты, соответствующие сменам состояния телеметрируемого объекта.

Далее представлен модифицированный алгоритм формирования символьного представления $\hat{C}$ для исходного временного ряда C .

Алгоритм 2. Построение символьного представления.

Вход: $C = c_1, \dots, c_n$ – исходный временной ряд; $P = p_1, \dots, p_w$ – w моментов смены состояния телеметрируемого объекта; $A = \alpha_1, \dots, \alpha_{k-1}$ – алфавит символьного представления длиной k ; $B = \beta_1, \dots, \beta_{k-1}$ – значения уровней. Выход: $\hat{C} = \hat{c}_1, \dots, \hat{c}_n$ – символьное представление.

1 $\hat{C} \leftarrow$ выполнить алгоритм 1 для C и P

2 $\bar{C}' \leftarrow$ нормализовать $\bar{C}$

3 для $i = 1$ до w

4 $\hat{c}_i = a_j \Leftrightarrow \beta_{j-1} \leq \bar{c}'_i < \beta_j$

5 конец для

Результат конкатенации символов $\hat{c}_1, \dots, \hat{c}_w$ определяет символьное представление временного ряда $\hat{C}$.

Редакционное расстояние и алгоритм его построения. Расстояние Левенштейна (также редакционное расстояние или дистанция редактирования) между двумя строками определяется как минимальное количество операций вставки / удаления одного символа или замены одного символа на другой, необходимых для преобразования одной строки в другую [14]. Под редакционным предписанием понимается последовательность действий, необходимых для преобразования первой строки во вторую с использованием минимального количества операций. При вычислении редакционного расстояния учитываются цены операций, которые зависят от вида операции (вставка, удаление, замена) и/или от участвующих в ней символов [15].

Для вычисления редакционного расстояния предлагается использовать алгоритм Вагнера-Фишера [16]. В алгоритме приняты следующие обозначения: $\omega(a, b)$ – цена замены символа a на символ b ; $\omega(e, b)$ – цена вставки символа b ;

$\omega(a, e)$ – цена удаления символа a . В качестве метрики расчета расстояния между символами рассматривается разность значений границ уровней соответствующих символов. Тогда функция расчета цены замены символа a на символ b имеет вид

$$\omega(a, b) = \begin{cases} 0, & \text{если } a = b, \\ |\beta_a - \beta_b|, & \text{если } a \neq b, \end{cases}$$

где β_a и β_b – границы уровней значений.

Алгоритм 3. Алгоритм Вагнера–Фишера.

Вход: $\hat{C}_1$, $\hat{C}_2$ – две сравниваемые строки, ω – функция, определяющая цену операций. Выход: D – редакционное расстояние между символическими представлениями $\hat{C}_1$, $\hat{C}_2$ временных рядов C_1 и C_2 соответственно.

```

1  $D(0, 0) = 0$ 
2 для  $j = 1$  до  $|\hat{C}_2|$ 
3  $D(0, j) = D(0, j-1) + \omega(e, \hat{C}_2[j])$ 
4 конец для
5 для  $i = 1$  до  $|\hat{C}_1|$ 
6  $D(i, 0) = D(i-1, 0) + \omega(\hat{C}_1[i], e)$ 
7 для  $j = 1$  до  $|\hat{C}_2|$ 
8  $D(i, j) = \min \begin{cases} D(i-1, j) + \omega(\hat{C}_1[i], e), \\ D(i, j-1) + \omega(e, \hat{C}_2[j]), \\ D(i-1, j-1) + \omega(\hat{C}_1[i], \hat{C}_2[j]). \end{cases}$ 
9 конец для
10 конец для
11 вернуть  $D(|\hat{C}_1|, |\hat{C}_2|)$ 
```

Оценки метода нечеткого сравнения телеметрических параметров. В методе построения символического представления ТМП для формирования РАА-представления необходимо просмотреть исходный временной ряд длиной n и вычислить среднее значение для ω сегментов, что тре-

бует временных затрат $O(n)$ и затрат по памяти $O(\omega)$. Перевод из РАА-представления в символическое представление осуществляется за линейное время и зависит от количества сегментов ω . Таким образом, суммарные временные затраты на формирование символического представления ТМП из исходного временного ряда составят $O(n + \omega)$. В связи с тем, что $n \gg \omega$, временные затраты, необходимые на выполнение метода, составляют $O(n)$.

Алгоритм Вагнера–Фишера, применяемый для сравнения временных рядов, в ходе работы осуществляет заполнение матрицы размером $|\hat{C}_1| \times |\hat{C}_2|$, что определяет время работы алгоритма и объем необходимой памяти, которые составляют $O(|\hat{C}_1| \cdot |\hat{C}_2|)$. Объем требуемой памяти может быть сокращен до линейного, если не сохранять промежуточные результаты работы алгоритма.

Разработанный метод приближенного сравнения телеметрических параметров на основе символического представления позволяет оценивать сходство анализируемых телеметрических параметров с ранее полученными параметрами от однотипных объектов, полученных при аналогичных условиях проведения ОИР. Реализация метода в системах интеллектуального анализа телеметрической информации позволила оперативно решать задачи оценки результатов проведенных ОИР. При обнаружении отклонений в поведении телеметрических параметров системой предоставляется информация о составе параметров, в которых выявлены отклонения, и моментах возникновения отклонений. Дальнейшее направление развития предложенного метода предполагает анализ возможности увеличения скорости работы метода за счет использования альтернативных алгоритмов вычисления редакционного расстояния между строками.

СПИСОК ЛИТЕРАТУРЫ

1. Назаров А. В. Современная телеметрия в теории и на практике: Учеб. курс. СПб.: Наука и техника, 2007.
2. Chan K., Fu A.W. Efficient Time Series Matching by Wavelets // In proc. of the 15th IEEE Int'l Conf. on Data Engineering, Sydney, Australia, Mar. 23–26, 1999. P. 126–133.
3. Faloutsos C., Ranganathan M., Manolopoulos Y. Fast Subsequence Matching in Time-Series Databases // In proc. of the ACM SIGMOD Int'l Conf. on Management of Data, Minneapolis, May 24–27, 1994. P. 419–429.
4. Geurts P. Pattern Extraction for Time Series Classification // In proc. of the 5th European Conf. on Principles of Data Mining and Knowledge Discovery, Freiburg, Germany, Sep. 3–7, 2001. P. 115–127.
5. Keogh E., Pazzani M. An Enhanced Representation of Time Series Which Allows Fast and Accurate Classification, Clustering and Relevance Feedback // In proc. of the 4th Int'l Conf. on Knowledge Discovery and Data Mining, N.Y., Aug 27–31, 1998. P. 239–241.

6. Scargle J., Jackson B., Norris J. Adaptive Piecewise-constant Modeling of Signals in Multidimensional Spaces // In proc. of the Statistical Problems in Particle Physics, Astrophysics and Cosmology, SLAC, Stanford, California, Sep. 8-11, 2003. P. 157-162.
7. Kawahara Y, Sugiyama M. Change-Point Detection in Time-Series Data by Direct Density-Ratio Estimation // In Proc. of the SIAM Intern. Conf. on Data Mining, SDM, April 30-May 2, 2009. Sparks.
8. Turner R., Saatei Y, Rasmussen C. E. Adaptive Sequential Bayesian Change Point Detection // Temporal Segmentation Workshop at NIPS. 2009. Dec. P. 1-4.
9. A Symbolic Representation of Time Series, with Implications for Streaming Algorithms / J. Lin, E. Keogh, S. Lonardi, B. Chiu // In proc. of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, San Diego, California. 13th June, 2003. P. 2-11.
10. Experiencing SAX: a Novel Symbolic Representation of Time Series / J. Lin, E. Keogh, S. Lonardi, B. Chiu // Data Mining and knowledge discovery. 2007. Vol. 15, № 2. P. 107-144.
11. Lin J., Keogh E., Fu. A.W. HOT SAX: Efficiently Finding the Most Unusual Time Series Subsequence // In proc. of the 5th IEEE Intern. Conf. on Data Mining, Houston, TX, Nov. 27-30, 2005. P. 226-233.
12. Keogh E., Chakrabarti K., Pazzani M. Locally Adaptive Dimensionality Reduction for Indexing Large Time Series Databases // In proc. of ACM SIGMOD Conf. on Management of Data, Santa Barbara, May 21-24, 2001. P. 151-162.
13. Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases / E. Keogh, K. Chakrabarti, M. Pazzani, S. Mehrotra // J. of Knowledge and Information Systems. 2001. Vol. 3. P. 263-286.
14. Левенштейн В. И. Двоичные коды с исправлением выпадений, вставок и замещений символов // Докл. АН СССР. 1965. С. 845-848.
15. Гасфилд Д. Строки, деревья и последовательности в алгоритмах: Информатика и вычислительная биология / Пер. с англ. И. В. Романовского. СПб.: Невский Диалект; БХВ-Петербург, 2003.
16. Wagner R. A., Fischer M. J. The string-to-string correction problem // J. of the ACM. 1974. Vol. 21, № 1. P. 168-173.

N. A. Zhukova, I. S. Sokolov, A. V. Ekalo

THE METHOD OF FUZZY COMPARISON OF TELEMETRIC PARAMETERS BASED ON DATA MINING

This paper offers the method of fuzzy comparison of slowly changing telemetric parameters based on symbolical representation with use of the weighed edit distance. Application of edit distance allows to compare parameters displaced from each other by time and/or values, and to eliminate insignificant deviations in parameters behavior, caused by influence of external factors.

Data Mining, symbolical representation, telemetric parameters

004.42.032.24

М. А. Фирсов, С. А. Ивановский

Параллельная реализация алгоритма построения пересечения простых полигонов с использованием технологии CUDA

Описана параллельная реализация на CUDA модифицированного расширенного алгоритма Грейнера-Хорманна построения пересечения простых плоских полигонов. Приведены результаты экспериментального сравнения времён выполнения последовательной версии и версии на CUDA.

Вычислительная геометрия, пересечение простых полигонов, технология CUDA

Операция построения оверлеев (объединения, пересечения или разности) плоских полигонов является одной из основных в системах автоматизированного проектирования (САПР), ГИС в других графических системах. На основе построения оверлеев решается целый ряд других смежных задач [1].

Разработано множество последовательных алгоритмов пересечения полигонов [2], но пока мало адаптаций этих алгоритмов для выполнения на графических ускорителях. В данной статье описывается параллельная версия для CUDA модифи-

Таблица 1

Ребро prev	Ребро next		
	out	on	in
out	enex	en	en
on	ex	none	en
in	ex	ex	exen

цированного расширенного алгоритма Грейнера–Хорманна построения пересечения двух простых плоских полигонов [3]. Этот алгоритм был выбран по следующим причинам:

- в [4] показана возможность успешного распараллеливания алгоритма пересечения множества полигонов (получено ускорение в 20...40 раз над последовательной версией для CPU);
- относительная простота алгоритма;
- основная работа 1-й фазы алгоритма Грейнера–Хорманна – поиск всех пересечений ребер полигонов – используется во многих других алгоритмах пересечения полигонов.

При разработке параллельной реализации использован подход, аналогичный описанному в [4]. При этом 1-я фаза алгоритма распараллелена практически так же, как в [4]; 2-я фаза – по-другому; 3-я, как и в [4], оставлена для последовательного исполнения на CPU, так как она вычислительно простая и последовательна по природе.

Модифицированный расширенный алгоритм Грейнера–Хорманна. Расширенный алгоритм Грейнера–Хорманна подробно описан в [3] и состоит из трех фаз.

Фаза 1. Нахождение всех точек пересечения ребер полигонов. Эти точки («вершины-пересечения») вставляются в соответствующие места списков вершин исходных полигонов и снабжаются флагом пересечения и указателем на вершину-двойника списка другого полигона.

Фаза 2. Производится обход полигона, и встреченные точки с флагом пересечения помечаются флагом входа (en), выхода (ex), входо-выхода (en-ex) либо выходо-входа (ex-en). Значение флага, сопоставляемое точке пересечения, определяется характеристиками предшествующего и следующего ребер (точнее, реберного фрагмента) для этой точки при обходе. Для ребра возможно 3 случая (3 типа принадлежности другому полигону):

- out – ребро лежит вне другого полигона;
- on – ребро совпадает с ребром другого полигона;
- in – ребро лежит внутри другого полигона.

В табл. 1 указано, какой флаг следует сопоставить точке пересечения в зависимости от характеристик предшествующего (prev) и последующего (next) ребер.

При определении принадлежности ребра тип on устанавливается просто (достаточно проверить, что его концы – вершины пересечения, а их двойники – соседи в списке вершин другого полигона). Если ребро не имеет тип on, то для определения принадлежности требуется проводить тест на попадание середины ребра внутрь другого полигона.

Фаза 3. Производятся обходы по спискам вершин исходных полигонов, дополненным вершинами-пересечениями, и получаются полигоны-пересечения.

Модифицированный расширенный алгоритм отличается от описанного в [3] некоторыми упрощениями, а также особенностями выполнения фазы 3.

Версия алгоритма для CUDA. Списки, использовавшиеся для представления полигонов в последовательном алгоритме, плохо подходят для обработки на GPU. С учётом особенностей доступа к глобальной памяти в CUDA для представления множества вершин используется набор одномерных массивов:

```
struct ArPolygon {
    COORDTYPE* xx;//массив x-координат
    COORDTYPE* yy;//массив y-координат
    UINT* an;//массив номеров (номерных значений)
    double* alfas;//массив значений alfa, характеризующих положение на ребре
    char* secflgs;//массив флагов
    UINT* dvoidniks;//массив индексов (указателей) вершин-двойников другого полигона
    UINT N;//число вершин в полигоне }.
```

В последовательности вершин i -я точка представляется совокупностью i -х элементов нескольких массивов. Так как полигон – замкнутая линия, полезно иметь копию вершины 0 в конце массива.

Изначально в массиве номеров содержатся последовательные числа $\{0, 1, 2, \dots, N-1, 0\}$, которые можно считать номерами соответствующих исходных ребер. Когда в полигон будут добавлены вершины-пересечения, то они получают номера

тех исходных рёбер, на которых располагаются (таким образом, в массиве номеров могут быть одинаковые числа). Место для вставки вершины-пересечения (рисунок) определяется по номерным значениям и значениям параметра $\alpha \in [0, 1]$.

Элемент массива флагов может принимать следующие значения:

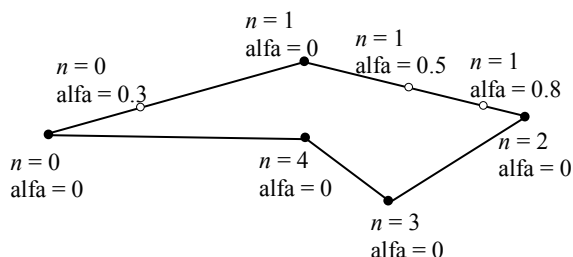
0 – вершина не является пересечением;

1 – вершина-пересечение с неизвестным флагом или с флагом none;

2, 3, 4, 5 – вершины-пересечения с флагами en, ex, enex и exen.

Совокупность массивов структуры `ArPolygon` назовем мультимассивом вершин.

Распараллеливание 1-й фазы алгоритма. Требуется найти все точки пересечения рёбер полигонов, установив у точек-двойников ссылки друг на друга. Точки пересечения разбивают исходные рёбра на рёберные фрагменты и становятся вершинами-пересечениями. Их следует добавить в полигоны, но добавлять вершины в массивы по отдельности крайне неэффективно. Поэтому в параллельной версии вершины-пересечения добавляются в полигоны не по одной, как в последовательной версии, а все вместе (после того, как все они будут найдены). При этом выполняется сортировка массива вершин, чтобы новые вершины заняли правильные места. Ключ сортировки – совокупность номерного значения и значения α (рисунок). Для сортировки использовалась самостоятельная реализация алгоритма быстрой сортировки на CUDA [5], при этом сначала упорядочивались ключи сортировки, а затем соответствующие перемещения выполнялись с самими вершинами.



Фаза 1 распараллелена в соответствии с идеей, изложенными в [4] (для алгоритма Грейнера–Хорманна). Пусть пересекаемые полигоны – А и Б. Тогда в фазе 1 выполняются следующие шаги:

1. Сделать копии исходных мультимассивов вершин. Эти копии понадобятся на следующей фазе 2.

2. Найти все пересечения рёбер и добавить их в мультимассивы полигонов А и Б.

3. Отсортировать мультимассив А.

4. Восстановить ссылки вершин мультимассива Б на вершины мультимассива А.

5. Отсортировать мультимассив Б.

6. Восстановить ссылки вершин мультимассива А на вершины мультимассива Б.

На шаге 2 используется метод двойного прохода – приём разработки параллельных алгоритмов, когда на первом проходе нити собирают информацию (каждая нить подсчитывает число найденных пересечений), а на втором – выполняют реальную работу: в мультимассивы добавляют пересечения (они находятся заново) в места, определённые с помощью первого прохода.

Распараллеливание 2-й фазы алгоритма. В [4] флаги вершин-пересечений предлагается устанавливать через выполнение префиксной суммы по вершинам-пересечениям, используя свойство чередования вершин-входов и вершин-выходов. В расширенной версии алгоритма это свойство может нарушаться, поэтому для определения флага вершины-пересечения требуется проверить принадлежность двух рёберных фрагментов к внутренней области или границе другого полигона (для этого лучше использовать его копию, сохранённую в начале фазы 1). Проверка принадлежности множества рёберных фрагментов – это работа, распараллеливаемая довольно очевидным образом, но всё же возникает немало вариантов. Далее приводятся особенности использовавшегося решения.

Каждой вершине-пересечению соответствует ровно один рёберный фрагмент, образованный этой вершиной-пересечением и её соседом справа (вершина является соседом справа, если расположена в массиве следующей). Определение принадлежности всех этих рёберных фрагментов требуется для расстановки флагов всем вершинам-пересечениям и составляет основную работу фазы 2. Флаги нужно проставить только вершинам-пересечениям, из остальных вершин будут использоваться лишь те, которые являются соседями справа для вершин-пересечений. Чтобы ненужные на фазе 2 вершины не мешали параллельной работе нитей, сначала составляется массив индексов (индексный массив) вершин-пересечений, во время прохода по которому и будет далее выполняться анализ рёбер, расстановка флагов.

Определение типов принадлежностей «OUT» и «IN» выполняется, как и в последовательной версии, с помощью алгоритма трассировки луча [6], применяемого к середине рёберного фрагмента и копии другого полигона. Каждая нить выполняет алгоритм трассировки луча для своего рёберного фрагмента, при этом нити, которым достались ON-рёбра, определяют это и алгоритм не выполняют, а лишь загружают данные для других нитей.

Пусть ставятся флаги пересечениям полигона А, при этом используется копия полигона Б; пусть в блоке имеются р нитей. Блок работает следующим образом:

1. Каждая нить проверяет, является ли соответствующее ей ребро ON-ребром.
2. Каждая нить загружает в разделяемую память координаты одной из вершин копии полигона Б.
3. Нити выполняют алгоритм трассировки луча, используя все загруженные вершины.
4. Если в копии полигона Б ещё остались вершины, не использовавшиеся в алгоритме трассировки луча, то переход на шаг 2 (повторяются шаги 2 и 3 для новой порции вершин копии полигона Б).
5. Нити обмениваются результатами для своих рёбер с соседними нитями, ставят флаг своей вершине-пересечению полигона А и, если возможно (см. далее), её двойнику.
6. Если в полигоне А ещё есть необработанные вершины пересечения, то переход на шаг 1 (повторяются шаги 1–6 для новой порции вершин-пересечений полигона А).
7. Установка флагов для крайних вершин-пересечений (подробности далее).

Для установки флагов крайних вершин-пересечений требуется информация (признак принадлежности левого инцидентного ребра) от соседнего блока. С помощью атомарных функций блоки синхронизируются со своими блоками-соседями, и те блоки, которые закончили работу раньше, оставляют информацию для тех блоков, которые закончат работу позже, чтобы последние поставили флаги для крайних вершин. Флаг крайней левой вершине устанавливает в итоге либо блок, которому она назначена, либо соседний слева блок. Для 0-го блока соседним слева является последний блок.

В описании шага 5 упоминается, что не всегда нить может поставить флаг двойнику обрабо-

танной вершины-пересечения. Если вершине-пересечению поставлен флаг ex или en, то двойник получает противоположный флаг, но если вершине-пересечению поставлен флаг exen или enex, то для определения флага двойника требуется дополнительная обработка. Для дополнительной обработки составляется индексный массив вершин полигона Б, которые всё ещё нуждаются в постановке флага, затем по индексному массиву выполняется их обработка. Она состоит в том, что нужно определить принадлежность (относительно копии полигона А) одного инцидентного ребра для каждой вершины, после чего возможно будет установить их флаги. Эта дополнительная обработка проще основной, так как на этот раз для установки флага не требуется информации от соседней нити или блока.

Результаты экспериментов.

Конфигурация тестовой системы:

Видеокарта: GeForce GTX 550 Ti 1024 MB.

Версия CUDA: 5.0.35.

Процессор: Intel Core i5-2400 3.1 GHz.

Компьютер: ОП – 3.49 Гб DDR3 1333 MHz (двухканальная), ОС – Windows XP SP3 32 bit.

В версии для CUDA время перемещения данных из ОП в глобальную память и обратно включалось в состав измеренного времени выполнения фазы 1 и 2 соответственно.

Измерено время пересечения таких пар многоугольников, в которых 2-й полигон получен смещением 1-го на вектор. В измерениях п. А этот вектор мал, чтобы было много пересечений рёбер; в измерениях п. В для одних и тех же полигонов этот вектор варьируется.

А. Исследование зависимости времён выполнения от количества вершин при большом числе пересечений рёбер.

Количество пересечений рёбер примерно равно половине суммарного количества вершин. Результаты измерений последовательной и параллельной версий алгоритма для разного количества вершин приведены в табл. 2.

Из таблицы можно сделать следующие выводы:

– параллельный алгоритм работает быстрее последовательного уже начиная с порядка 1000 вершин;

– в данном случае 1-я фаза отнимает около 94 % времени в последовательном алгоритме и около 96 % времени в параллельном;

– параллельный алгоритм в целом достигает ускорения в 12,7 раза, по фазе 1 – в 12,3 раза, а по фазе 2 – в 24 раза.

В. Исследование зависимости времён выполнения от количества пересечений рёбер при большом числе вершин.

Использован полигон с 33 378 вершинами. Из результатов измерений времён выполнения последовательной и параллельной версий алгоритма для разного количества пересечений рёбер можно сделать следующие выводы:

– число пересечений практически не влияет на длительность фазы 1 в последовательном алгоритме и влияет в параллельном алгоритме;

– длительность фазы 2 линейно зависит от числа пересечений, как и должно быть.

Разработанный модифицированный расширенный алгоритм Грейнера–Хорманна для CUDA демонстрирует значительное ускорение над своим последовательным аналогом (до 13 раз) и может быть использован (после простых модификаций) при решении задач, требующих пересечения простых полигонов с большим количеством вершин.

В дальнейшем представляется целесообразным: оптимизация алгоритма с учётом более тонких особенностей CUDA; исследование эффективности других вариантов реализации модифи-

Таблица 2

Количество вершин в 1-м полигоне	Число пересечений	Последовательный алгоритм (мс)			Алгоритм для CUDA (мс)		
		Фаза 1	Фаза 2	Фазы 1 и 2	Фаза 1	Фаза 2	Фазы 1 и 2
340	347	0,005	0,000	0,005	0,023	0,001	0,024
564	575	0,014	0,001	0,015	0,025	0,001	0,026
1176	1110	0,058	0,003	0,061	0,035	0,002	0,037
1828	1825	0,140	0,009	0,149	0,036	0,002	0,038
3474	3350	0,50	0,03	0,53	0,066	0,004	0,070
6120	6163	1,55	0,11	1,66	0,151	0,008	0,159
10132	10092	4,2	0,3	4,5	0,38	0,02	0,40
18224	18222	13,8	0,9	14,7	1,12	0,04	1,16
33378	34513	45,4	3,3	48,7	3,77	0,14	3,91
55372	57218	–	–	–	10,0	0,4	10,4

– число пересечений сильно влияет на длительность фазы 2;

– несмотря на то, что фаза 2 распараллелена лучше фазы 1 (ускорение фазы 2 – до 32 раз), параллельный алгоритм оказывается сравнительно эффективнее при меньшем числе пересечений (достигает ускорения в 13 раз), так как при этом уменьшается длительность его фазы 1 (она достигает ускорения в 13 раз) (неизвестно, сохранится ли эта тенденция при увеличении числа пересечений до порядка квадрата от числа вершин);

цированного расширенного алгоритма Грейнера–Хорманна для CUDA (исследована эффективность лишь одного из возможных вариантов реализации), в том числе использования эффективных алгоритмов поиска всех пересечений рёбер; обработка особых случаев в модифицированном расширенном алгоритме Грейнера–Хорманна, связанных с ограниченной точностью чисел типа double; дальнейшие исследования влияния характеристик входных данных на время работы вариантов модифицированного расширенного алгоритма Грейнера–Хорманна на CUDA.

СПИСОК ЛИТЕРАТУРЫ

1. Скворцов А. В. Построение объединения, пересечения и разности произвольных многоугольников в среднем за линейное время с помощью триангуляции // Вычислительные методы и программирование. 2002. Т. 3. С. 116–123.

2. Ченцов О. В., Скворцов А. В. Обзор алгоритмов построения оверлеев многоугольников // Вестн. Томского гос. ун-та. 2003. № 280. С. 338–345.

3. Dae Hyun Kim. An Extension of Polygon Clipping To Resolve Degenerate Cases // Computer-Aided Design & Applications. 2006. Vol. 3, № 1–4. P. 447–456.

4. Jianting Zhang. Polygon Overlay Operations in CudaGIS (Initial Design & Implementation) / Jianting

Zhang, Simin You, Kai Zhao. – Режим доступа: http://www-cs.ccnycunyu.edu/~jzhang/papers/PolyOvr_dr.pdf

5. Cederman D., Tsigas Ph. GPU-Quicksort: A Practical Quicksort Algorithm for Graphics Processors // ACM J. of Experimental Algorithmics (JEA). 2009. Vol. 14, Dec (секция 1, статья № 4).

6. Препарата Ф., Шеймос М. Вычислительная геометрия. М.: Мир, 1989. (См. также: Point in polygon // Wikipedia. – Режим доступа: http://en.wikipedia.org/wiki/Point_in_polygon#Ray_casting_algorithm.)

M. A. Firsov, S. A. Ivanovskiy

PARALLEL IMPLEMENTATION OF ALGORITHM FOR CONSTRUCTION OF SIMPLE PLANE POLYGON INTERSECTION WITH USE OF CUDA TECHNOLOGY

Parallel implementation of modified extended Greiner-Hormann algorithm for construction of simple plane polygon intersection with use of CUDA is described. Experimental comparison results of sequential version and version for CUDA run times are given.

Computational geometry, simple polygon intersection, CUDA



УДК 004.9:004.657

Н. М. Литвиненко, С. В. Родионов

Универсальный конструктор структур данных для систем каталогизации информации

Приводится информация об универсальном конструкторе структур данных для систем каталогизации информации, созданном в рамках Научно-технической программы Союзного государства «Разработка и создание нового поколения микросистемотехники и унифицированных интегрированных систем двойного назначения на их основе» на 2010–2013 гг. Обсуждаются преимущества универсального конструктора, позволяющие использовать его в качестве полноценной, индивидуально настраиваемой системы хранения информации. Особо отмечается возможность создания коробочной версии универсального конструктора для предприятий и организаций любых сфер и направлений деятельности, что подтверждается результатами его апробации и внедрения в ОАО «Авангард».

Универсальный конструктор структур данных, база данных, система каталогизации, хранение и поиск данных, интерфейс пользователя, предметная область

Современные экономические условия диктуют необходимость разработки гибкого, динамичного, индивидуально настраиваемого программного продукта, который предоставит предприятиям и организациям любой отрасли промышленности возможность самим создавать системы каталогизации продукции, товаров и услуг.

Производство деталей и узлов сложных технических объектов требует в настоящих условиях обязательного учета требований каталогизации продукции в соответствии с принципами, принятыми в Федеральной системе каталогизации продукции для федеральных государственных нужд (ФСКП) и международной практике каталогизации продукции.

Системы каталогизации товаров и услуг одинаково востребованы как крупными производственными предприятиями, для которых большое значение имеет четкая регламентация бизнес-процессов, так и предприятиями среднего и малого бизнеса, которым необходимо грамотно проводить ассортиментную политику.

Конструктор структур данных как универсальная настраиваемая система классификации и хранения информации может быть внедрен на предприятиях и организациях любой области промышленности.

На сегодняшний день на рынке программного обеспечения представлено множество систем каталогизации продукции. Среди наиболее популярных можно назвать Content Management System (CMS), Ontology Web Language (OWL), Triplets, Resource Description Framework (RDF). Несмотря на явные достоинства вышеперечисленных систем все они имеют 2 существенных недостатка – требуют специальных знаний для их применения и не предоставляют возможности гибко изменять структуру данных. Данные системы рассчитаны на специалистов в области информационных технологий, специальные знания и навыки которых позволяют им работать с данными программными продуктами. Однако для специалистов предметной области, которые являются основными пользователями систем каталогизации продукции, использование в повседневной деятельности данных программных продуктов вызвало бы затруднения ввиду отсутствия у них пользовательской части с доступным интерфейсом, посредством которого можно было бы оперировать именно понятиями предметной области.

В процессе анализа рынка стала очевидной необходимость разработки гибкого, динамичного, индивидуально настраиваемого программного

продукта, который предоставит предприятиям и организациям любой отрасли промышленности возможность самим создавать системы каталогизации продукции, товаров и услуг.

Возможность внедрения представленной идеи, в частности, подтверждается необходимостью решений по комплексной информационно-аналитической поддержке жизненного цикла нового поколения изделий микросистемотехники (ИМСТ), разрабатываемых ОАО «Авангард» (Санкт-Петербург) в рамках российско-белорусской научно-технической программы Союзного государства, в части процесса формирования электронного каталога, содержащего информацию об ИМСТ, включая конструкторско-технологическую документацию, материалы и технологии производства. В рамках данного проекта проводилось проектирование, разработка, внедрение и апробация модели гибкого каталога, после чего был изготовлен опытный образец, который предусматривает возможность создания коробочной версии программного продукта.

В основу проектирования универсального конструктора положены принципы трехуровневой системы каталогизации, позволяющие описывать практически любую предметную область.

Создание структур данных опирается на следующую иерархию объектов [1], [2]:

- категории;
- группы объектов;
- группы характеристик.

При этом обеспечивается возможность задания множественных условий для значений характеристик, а также возможность в качестве значений характеристик задавать имена прикрепленных файлов (документы, фотографии, схемы, таблицы, диаграммы, 3D-объекты, звукозаписи, видеофрагменты и т. п.) и ссылки на другие связанные объекты.

Этапами проектирования и разработки универсального конструктора можно назвать:

- проведение комплексного анализа систем каталогизации, применяемых на предприятиях Санкт-Петербурга и Северо-Западного региона;
- выявление недостатков существующих систем каталогизации и выработку предложений по их улучшению;
- изучение возможностей построения и технологий создания систем каталогизации с настраиваемой структурой хранения данных;
- проектирование и создание базы данных, способной хранить настраиваемую структуру каталога;

– разработку пользовательского интерфейса конструктора структур данных для проекта системы каталогизации с настраиваемой структурой;

– разработку пользовательского интерфейса редактора ввода данных для проекта системы каталогизации с настраиваемой структурой.

Универсальный конструктор структур данных может рассматриваться с двух точек зрения. С одной стороны – это система каталогизации, представляющая собой организационно упорядоченную совокупность информации о разнородных объектах (каталог продукции, технологий, документации). С другой стороны – это совокупность информационных технологий, обеспечивающих представление, сбор, классификацию, идентификацию, кодирование, регистрацию, обработку, хранение и распределение информации, которая необходима для реализации производственных процессов.

Универсальный конструктор структур данных обладает целым рядом преимуществ:

• *Гибкость.* Основным принципом, который был положен в основу проектирования универсального конструктора, – возможность его использования для классификации любого вида продукции и, как следствие, возможность его самого широкого применения на рынке.

• *Адаптивность.* Возможность изменять структуру данных в ходе эксплуатации в зависимости от текущих потребностей предприятия. По сути это означает возможность добавления новых объектов описания с новыми наборами характеристик.

• *Лояльность к измерительным шкалам.* Конвертор единиц измерения позволяет сводить воедино данные о разнородных по своим параметрам типах и видах продукции.

• *Лояльность к пользователю.* Для внедрения, настройки и эксплуатации представляемой системы каталогизации не требуется специальных знаний и навыков. Пользовательский интерфейс системы каталогизации предоставляет удобный доступ к структуре данных.

В качестве основных положительных потребительских свойств представляемого программного продукта можно назвать следующие свойства:

- визуальное описание понятий предметной области для последующего структурированного ввода данных о произвольной номенклатуре товаров и услуг;
- проверку корректности вводимых данных;
- возможность согласования вводимых данных в соответствии с бизнес-процессами, принятыми у пользователя;

- широкие возможности по поиску данных;
 - расширяемый справочник единиц измерения;
 - возможность хранения прикрепленных файлов различных форматов (текстовые документы, изображения, схемы и чертежи, медиафайлы);
 - разграничение прав доступа к функционалу на основе ролей;
 - возможность интеграции с внешними системами;
 - защиту от несанкционированного доступа.
- Конкурентные преимущества:
- реализация в виде клиент-серверного web-приложения предоставляет возможность совместной работы многих пользователей, имеющих доступ в Интернет;
 - не требуется установка специального программного обеспечения для работы клиентской части;
 - настройка структур хранения данных под предметную область пользователя осуществляется через удобный интерфейс;
 - настраиваемые процессы согласования ввода данных позволяют контролировать сохранность и корректность информации;
 - минимальные требования к техническим средствам для работы серверной части (необходима СУБД MS SQL 2008 R2);
 - возможность контроля вводимой информации экспертом в режиме согласования;

- гибкие режимы создания прав пользователей для предоставления конфиденциальной и секретной информации;
- возможность интеграции с другими системами посредством передачи данных через web-сервисы;
- реализация на кроссплатформенных технологиях (MCBC, БД Линтер BC, Tomcat, Java, AJAX-интерфейс Web 2.0).

Универсальный конструктор позволяет осуществлять комплексный поиск данных об объектах, информация о которых занесена в систему.

Реализованы следующие механизмы поиска:

- поиск информационных объектов по значениям их характеристик;
- поиск значений характеристик для заданной группы объектов;
- поиск объектов, аналогичных выбранному по указанным характеристикам.

Помимо этого реализована подсистема защиты доступа к функциям информационной среды на основе ролей, защищенных именем и паролем пользователя.

Все изменения структур данных проходят цепочки согласований, тем самым реализуется аналог бизнес-процесса принятия решений по изменению информационных структур.

СПИСОК ЛИТЕРАТУРЫ

1. Порядок формирования, ведения и использования разделов федерального каталога продукции для федеральных государственных нужд: Методические рекомендации / В. М. Буренок, А. В. Губанов, Е. А. Давыдов и др. М., 2003.
2. ГОСТ Р 51725.1–2002. Каталогизация продукции для федеральных государственных нужд. Федеральная система каталогизации продукции для федеральных государственных нужд. Основные положения. М.: Изд-во стандартов, 2002.

N. M. Litvinenko, S. V. Rodionov

DATA STRUCTURE BUILDER FOR INFORMATION CATALOGING SYSTEMS

The article deals with data structure builder for information cataloging systems. It gives a detailed analysis of superiority of data structure builder as compared with other analogs. Its main preference is possibility of creating of the «boxed version» for commercial organization. data structure builder had introduced and tested at the customer.

Data structure builder for information cataloging systems, database, information cataloging system, CRUD-based systems, user interface, knowledge domain

УДК 681.32

В. А. Кирьянчиков, А. С. Романов

Автоматическая генерация тестов объектно-ориентированных программ по диаграммам деятельности

Рассматривается методика автоматической генерации интеграционных тестов на основе диаграмм деятельности языка UML, приводится пример диаграммы деятельности с различными типами действий и ограничений на их выполнение, предлагается алгоритм создания сценария тестирования на основе заданных в диаграмме действий и ограничений, приводится сформированный по диаграмме набор тестов, обеспечивающих проверку корректности работы моделируемой программы.

Тестирование на основе моделей, язык UML, диаграмма деятельности, генерация интеграционных тестов, язык OCL, система ПРОЛОГ

Данная статья является развитием методологии тестирования программ на основе моделей языка UML, примененной в работах [1], [2] для создания методики автоматизированной генерации модульных и интеграционных тестов. Если методика генерации модульных тестов в [1] была основана на использовании диаграмм состояний, а в работе [2] предложена генерация интеграционных тестов на основе диаграмм последовательности, то в настоящей статье предлагается методика формирования интеграционных тестов на основе диаграмм деятельности, моделирующих поведение программы в виде последовательности действий, выполняемых ее различными объектами.

Диаграммы деятельности (ДД) используются для моделирования динамических аспектов системы. Диаграммы этого класса похожи на диаграммы состояний (ДС), поскольку оба вида диаграмм представляют последовательность состояний системы во времени и условия, вызывающие переход из одного состояния в другое. Различие между этими диаграммами состоит в том, что ДС чаще используются для моделирования поведения одного объекта, тогда как ДД используются для моделирования некоторого процесса, в котором участвуют несколько объектов [3]. Кроме того, на основе ДД удобнее моделировать различные управляющие конструкции программ, такие, как ветвления и циклы. В языке UML также имеются средства для моделирования на основе ДД параллельных процессов, но в данной статье эти средства не рассматриваются. Ввиду того, что ДД используются для моделирования процессов,

в которых могут принимать участие несколько объектов, сгенерированные на основе этих диаграмм тесты могут быть использованы для интеграционного тестирования. Особенно удобно их применять при регрессионном тестировании, так как при изменении кода какого-либо класса можно определить, на каких диаграммах присутствуют объекты этого класса, а следовательно, выбрать тесты, которые необходимо выполнить для проверки работоспособности системы после внесения изменений.

Основными элементами диаграмм деятельности, графически представляемых в виде ориентированных графов, являются узлы-деятельности. С помощью этих узлов моделируются какие-либо действия, арифметические вычисления или вызовы методов объекта в программе. Также узлами-деятельности представляются такие элементы потока управления, как распараллеливание (fork), объединение (join), ветвление (decision) и слияние (merge). Наконец, на ДД присутствуют начальный и конечный узлы, в которых соответственно начинается и заканчивается моделируемый процесс. На ДД узлы-деятельности, моделирующие действия, изображаются в виде прямоугольников со скругленными углами; узлы, моделирующие ветвление и слияние, изображаются в виде ромбов; узлы, моделирующие распараллеливание и объединение, изображаются с помощью жирных линий. Начальное состояние на ДД показано в виде закрашенного круга, а конечное состояние изображается в виде закрашенного круга внутри окружности.

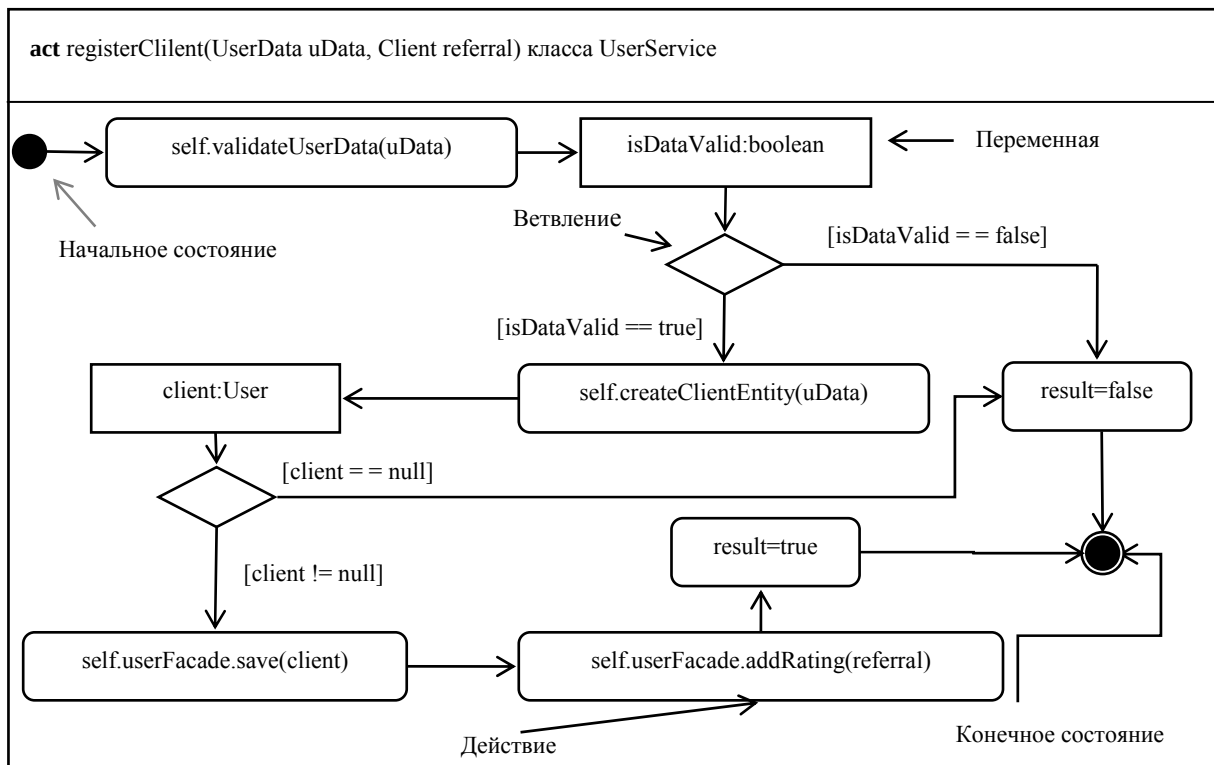
Другим типом узлов, присутствующих на ДД, являются узлы-переменные. Эти узлы моделируют входные и/или выходные параметры узлов, моделирующих действие, и по существу представляют поток данных, обрабатываемых в программе. Узлы-переменные изображаются в виде обычных прямоугольников с прямыми углами.

Все узлы диаграммы деятельности связаны между собой направленными дугами, показывающими, в какой последовательности выполняются действия и какие объекты при этом порождаются и используются. Дуги могут иметь защитные условия, которые обязаны быть выполнены (принять значение «истина»), чтобы переход по дуге был осуществлен. Защитными условиями обычно нагружаются дуги, которые выходят из узлов-ветвлений.

Пример диаграммы деятельности, моделирующей процесс регистрации клиента в некоторой информационной системе, приведен на рисунке. В системе наряду с обычной регистрацией пользователей предусмотрена возможность регистрации клиентов на основе использования реферальных ссылок. При старте процесса регистрации система находится в начальном состоянии, из которого она переходит к действию `self.validateUserData(uData)`. Это действие заключается в проверке введенных пользователем данных, называемой валидацией.

По окончании этого действия результат проверки записывается в булевскую переменную `isDataValid`. Узел-ветвление разделяет маршрут дальнейшего выполнения регистрации на 2 пути. Если значение переменной `isDataValid` равно `false`, то осуществляется переход в конечное состояние и процесс регистрации завершается. Если значение переменной `isDataValid` равно `true`, то выполняется действие `self.createClientEntity(uData)`, при котором создается объект `client`, представляющий запись клиента в таблице БД. Далее осуществляется сохранение этого объекта в БД посредством выполнения действия `self.userFacade.save(client)`. Заключительным является действие `self.userFacade.addRating(referral)`, при котором рефералу начисляется рейтинг.

По аналогии с моделями на основе диаграмм состояний и диаграмм последовательности, рассмотренными в [1], [2], модель в виде диаграммы деятельности также может быть детализирована ограничениями на языке Object Constraint Language (OCL) [4]. Эти ограничения могут отражать пред- и постусловия операций, которые представлены условиями выполнения действий, или актуальные значения переменных, присутствующих на диаграмме. Кроме того, OCL-ограничениями могут быть нагружены дуги, выходящие из узлов-ветвлений.



ОСЛ-ограничения – это выражения на языке объектных ограничений OCL, аргументами которых являются элементы модели, а результат может быть равен true или false. Если ограничение принимает значение true, то говорят, что оно выполнено, если false – то не выполнено. Для того чтобы детализировать ДД ОСЛ-ограничениями, необходимо иметь диаграмму классов, на которой будут смоделированы все классы, объекты которых участвуют в деятельности, моделируемой ДД. На этой диаграмме классов определяются инварианты классов, пред- и постусловия методов, которые также должны быть записаны на языке OCL.

Для проверки правильности выполнения какого-либо действия на ДД необходимо решить 3 задачи. Во-первых, привести систему в такое состояние, при котором будут выполнены инварианты объектов, участвующих в этом действии, и выполнено предусловие метода, представленного действием. Во-вторых, необходимо выполнить само действие. И в-третьих, необходимо проверить, что не нарушены инварианты объектов и выполнено постусловие метода. Как уже отмечалось, инварианты, пред- и постусловия могут быть записаны на языке OCL. Чтобы привести систему в нужное состояние, необходимо решить булевское ОСЛ-выражение. Под решением подразумевается поиск таких значений переменных, при которых данное выражение принимает значение true. Поиск решений булевского выражения можно осуществить с помощью системы ПРОЛОГ. Для этого необходимо выражение на языке OCL транслировать в выражения языка ПРОЛОГ и запустить их выполнение с целью поиска подходящих значений переменных. Данный подход уже был использован при описании методики генерации модульных тестов по диаграммам состояний [1].

Особенность интеграционного тестирования состоит в том, что тестируется не одиночный объект, а система взаимодействующих объектов. При этом часто один тест проверяет не одиночное действие, а последовательность действий. Если какое-либо действие из этой последовательности выполняется некорректно, то правильность остальных действий не проверяется и тест считается не пройденным. Использование ДД для генерации интеграционных тестов позволяет определить все возможные маршруты выполнения программы. Для каждого маршрута можно составить последовательность выполняемых действий,

которая будет основой интеграционного теста, т. е. теста, проверяющего корректность работы системы по соответствующему маршруту выполнения.

Для поиска всех маршрутов выполнения исходную диаграмму деятельности следует преобразовать в ориентированный граф следующим образом:

- каждый узел ДД заменить на вершину графа;
- каждый переход заменить на дугу графа, соединяющую вершины, соответствующие конечным узлам перехода.

Полученный в результате такого преобразования граф будет полностью повторять по структуре исходную ДД и содержать вершины, соответствующие начальному и конечному состояниям ДД, которые обозначим v_s и v_e соответственно. Для поиска возможных маршрутов выполнения программы между вершинами v_s и v_e по ее графу в литературе (например, в [5]) описаны формализованные методы, позволяющие автоматизировать процесс их получения. В этих методах выбор используемых маршрутов производится по различным критериям (минимального покрытия, цикломатического числа и т. д.) в зависимости от требований к точности тестирования.

После определения набора проверяемых маршрутов для подготовки тестов необходимо выполнить обратное преобразование маршрутов на графе в маршруты, представляющие последовательность действий в программе, выраженную в терминах ДД, следующим образом:

- каждую вершину заменить на соответствующий узел ДД;
- каждую дугу заменить на соответствующий переход между узлами ДД.

Далее разработка тестов для проверки правильности выполнения каждого маршрута складывается из следующих шагов:

- 1) инициализация значений входных данных;
- 2) выполнение действия текущего узла маршрута, если он не конечный, иначе – выход;
- 3) проверка постусловия и переход на п. 2, если результат проверки true, иначе – завершение тестирования.

Если все проверки постусловий выполнены успешно, то тест можно считать пройденным.

Наиболее сложной проблемой данной методики является инициализация входных данных в соответствии с начальными условиями. Следует отметить, что эта задача не всегда может быть решена. Возможность решения зависит от полно-

ты UML-модели. Однако существующие подходы к написанию интеграционных тестов позволяют обойти это ограничение, например используя заглушки вместо объектов реальных классов.

Можно выделить 2 вида данных, которые используются в программе при выполнении определенного маршрута:

1) входные данные, которые поставляются объектом класса;

2) порожденные данные, т. е. данные, полученные в результате выполнения действия.

Так, для изображенной на рисунке модели входными данными являются объекты `uData` класса `UserData` и `referral` класса `Client`, а порожденным данным является значение переменной `isDataValid` типа `boolean`. Это значение является результатом выполнения метода `validateUserData(uData)`. Язык OCL позволяет на уровне модели описать результирующее значение метода с помощью секции `def*`, что дает возможность подготовить начальные данные таким образом, чтобы получить требуемые значения порожденных данных. Например, для модели, изображенной на рисунке, секция `def` метода `validateUserData(uData:UserData)` может быть определена следующим образом:

```
def: validateUserData(uData : UserData) : boolean
= (false = uData.firstName.ocllsUndefined() and
false = uData.lastName.ocllsUndefined())
```

Таким образом, результирующее значение метода `validateUserData` будет равно `true`, если данные о пользователе содержат его имя и фамилию. Это определение не только позволяет проверять объекты класса `UserData`, но также задает правило создания валидных и невалидных объектов. Иными словами, это выражение можно использовать для того, чтобы сгенерировать требуемый объект `uData`, который поступает на вход функции `registerClient`.

Для того чтобы подготовить входные данные для тестирования определенного маршрута, необходимо найти решение OCL-ограничения, которое является конъюнкцией следующих более мелких ограничений:

1) инварианты всех классов, объекты которых участвуют в тестируемом взаимодействии;

2) предусловия всех методов, которые будут вызваны в процессе выполнения теста;

3) предусловие моделируемого взаимодействия;

4) для каждого конкретного значения порожденного данного должно выполняться условие получения этого значения.

Областью решения этого OCL-ограничения будет набор конкретных значений или интервалов возможных значений каждой переменной, используемых в этом выражении. Каждый набор конкретных значений всех переменных будет удовлетворять предусловию теста.

Как уже отмечалось, после инициализации маршрута входными значениями необходимо выполнить каждый метод из тестируемого маршрута и проверить постусловие выполненного метода. Если в результате все постусловия будут выполнены, то данный маршрут пройден успешно. Для модели, изображенной на рисунке, тест будет проверять правильность выполнения следующих маршрутов:

1) `self.validateUserData(uData), result=false;`

2) `self.validateUserData(uData),`
`self.createClientEntity(uData); result=false;`

3) `self.validateUserData(uData),`
`self.createClientEntity(uData),`
`self.userFacade.save(client),`
`self.userFacade.addRating(referral); result= true.`

Для простоты и ясности примера будем использовать только проверки на `null` и операцию сравнения чисел. Инвариантом класса `UserService` является выражение `self.userService.ocllsUndefined()=false`, означающее, что объект `self.userService` определен и его значение не `null`. Предусловием метода `UserService.registerClient` является выражение `uData.ocllsUndefined=false and referral.ocllsUndefined()`. Предусловия методов, записанные на языке OCL, приведены в таблице.

Метод	Предусловие на языке OCL
<code>UserService.validateUserData(uData)</code>	<code>uData.ocllsUndefined() = false</code> Это условие покрывается предусловием всего моделируемого взаимодействия
<code>UserService.createClientEntity(uData)</code>	<code>uData.ocllsUndefined() = false and</code> <code>uData.firstName.ocllsUndefined() =</code> <code>= false and uData.lastName</code> <code>.ocllsUndefined() = false</code>
<code>userFacade.save(client)</code>	<code>client.ocllsUndefined()=false</code>
<code>UserFacade.addRating(referral)</code>	<code>referral.ocllsUndefined()=false</code> Это условие покрывается предусловием всего моделируемого взаимодействия

Постусловием метода `UserFacade.addRating(referral)` является выражение `referral.rating > re-`

* Object Management Group, Object Constraint Language version 2.3.1. <http://www.omg.org/spec/OCL/2.3.1>.

ferral.rating@pre, показывающее, что значение поля referral.rating после выполнения метода больше, чем до его выполнения.

Для первого маршрута требуется тест, проверяющий значение переменной result при условии, что объект uData некорректный, например атрибут name этого объекта равен null. В результате выполнения теста значение result должно быть равно false.

Для второго маршрута нужно составить тест, который проверит значение переменной result при условии, что объект client равен null. В результате выполнения теста значение result также должно быть равно false.

Для третьего маршрута нужно составить тест, который проверяет значение переменной result при выполнении всех пред- и постусловий. В результате выполнения теста значение переменной result должно быть равно true.

Следует обратить внимание, что для маршрутов 2 и 3 объект client создается в результате работы объекта UserService.userFacade. При этом в двух тестах необходимо иметь разные значения объекта client при наличии одинакового входного условия. Чтобы провести полноценное тестирование класса UserService, можно использовать 2 разных mock-объекта, т. е. 2 фиктивных объекта, имитирующих работу объекта реального класса

исключительно для целей тестирования. Первый mock-объект будет возвращать в результат выполнения метода createClientEntry(UserData) значение null, а второй – любой объект класса Client, но не null.

В заключение хотелось бы отметить, что диаграммы деятельности языка UML в совокупности с диаграммами классов позволяют моделировать как структуру системы, так и ее поведение, а мощность языка позволяет применять его к моделированию систем любой сложности и уровня абстракции. В свою очередь язык OCL, предназначенный для записи объектных ограничений, является хорошо формализованным языком, позволяющим широко известными методами [6] транслировать записанные на нем выражения в программы на другом языке, в том числе и на языке ПРОЛОГ. Это позволяет автоматически получать решения OCL-выражений и использовать их для подготовки входных данных при генерации тестов для проверки требуемых маршрутов. Следовательно, можно утверждать, что совместное использование языков UML и OCL позволяет создавать модели программ, с помощью которых можно средствами автоматизации генерировать наборы тестов, проверяющих правильность работы моделируемых программ.

СПИСОК ЛИТЕРАТУРЫ

1. Кирьянчиков В. А., Романов А. С. Автоматизированная генерация модульных тестов по диаграммам состояний // Изв. СПбГЭТУ «ЛЭТИ». 2011. Вып. 9. С. 55–60.
2. Кирьянчиков В. А., Романов А. С. Автоматизация генерации тестов программ по диаграммам последовательности // Изв. СПбГЭТУ «ЛЭТИ». 2012. Вып. 9. С. 53–59.
3. Буч Г., Рамбо Д., Джекобсон А. Язык UML. Руководство пользователя. СПб.: ДМК-Пресс, 2004.
4. Warmer J., Kleppe A. Object Constraint Language, The: Getting Your Models Ready for MDA. Second Edition. Addison Wesley, 2003.
5. Липаев В. В. Качество программного обеспечения. М.: Финансы и статистика, 1983.
6. Опалева Э. А., Самойленко В. П. Языки программирования и методы трансляции. СПб.: БХВ-Петербург, 2005.

V. A. Kiryanchikov, A. S. Romanov

THE AUTOMATED OBJECT-ORIENTED PROGRAMS TEST GENERATION BASED ON THE ACTIVITY DIAGRAMS

The article describes technique of the automated integration test generation based on UML activity diagrams. An example of the activity diagram is considered and the method of the testing scenario development is suggested. In conclusion the appropriate sets of test descriptions are given.

Model-Based Testing, UML, activity diagram, generation of integration tests, OCL, Prolog

УДК 004.94, 004.85

Е. Л. Калишенко, Ю. В. Романенко

Технологические аспекты реализации компьютерных обучающих систем с трехмерной графикой в среде QNX

Описываются технологические аспекты разработки трехмерных графических приложений в составе компьютерных обучающих систем. Рассматриваются различные варианты оптимизации и повышения производительности работы графических приложений в среде QNX.

Компьютерное обучение, тренажеры, трехмерные графические приложения, операционная система QNX, оптимизация работы графических приложений

В современных экономических условиях, когда даже мелкий ремонт сложного технологического оборудования сложных технических комплексов грозит превратиться в неразрешимую проблему, цена ошибок эксплуатирующего персонала многократно возрастает. Особую важность приобретает качественное обучение и постоянное поддержание квалификации и готовности персонала, его противоаварийные тренировки. Актуальной является проблема создания адекватной системы обучения и аттестации, способной быстро реагировать на изменяющиеся требования и условия эксплуатации сложных систем, устройств, механизмов.

В последние годы в практике инженерной подготовки все чаще используют компьютерные обучающие системы (КОС). КОС представляют собой аппаратно-программные комплексы, которые легко перенастраиваются при модернизации оборудования, позволяют построить обучение на основе проблемно-ситуационного подхода, реализовать активные (интерактивные) методики подготовки. КОС создают возможность реализовать практически любые по сложности эксперименты с оборудованием и воспроизвести методики отработки любых штатных и нештатных ситуаций, не используя реальные устройства и агрегаты.

Современные КОС по назначению и функциональным возможностям условно разделяются на следующие группы:

– *компьютерные учебные пособия* – программно-методические комплексы, обеспечивающие возможность самостоятельно освоить учебный курс, соединяющие в себе свойства обычного учебника, справочника, задачника. Обычно ком-

пьютерные учебные пособия выполняются в виде электронных презентаций, учебных фильмов и интерактивных гипертекстовых баз знаний и т. д.;

– *интерактивные обучающие комплексы* – это пакеты программ, позволяющие оператору самостоятельно изучить учебный материал в интерактивном режиме и провести самопроверку;

– *компьютерные тренажеры* – аппаратно-программные комплексы, реализующие формализованную модель изменения состояния систем управления техническими комплексами в зависимости от действий операторов и задаваемых внешних факторов. Обучаемый взаимодействует с виртуальными устройствами, внешний вид и технология работы с которыми повторяют реальное оборудование. Такие тренажеры можно использовать для индивидуального и группового обучения, моделирования штатных технологических процессов и нештатных ситуаций, отработки и закрепления практических навыков совместной работы с оборудованием. Они обеспечивают возможность обучения и тренировки на различных уровнях самостоятельности обучаемого, контроль действий операторов.

Одной из важнейших функций КОС, определяющей качество и результативность обучения эксплуатирующего персонала, является отображение объектов реального мира. Реализация этой функции связана с воспроизведением визуальных образов с использованием мультимедиа-технологий, многовариантной системы моделирования, имитацией управления объектами путем интерактивного взаимодействия. Наглядность представления и удобство работы оператора с КОС дости-

гаются за счет современных компьютерных технологий, включая видеоизображения, интерактивные двухмерные и трехмерные модели с возможностью изменения параметров отображения и обзора модели, анимационные ролики. Визуальные образы объектов в большинстве современных КОС создаются с помощью программно-генерируемой графики. Для реализации движений, динамических процессов или физических явлений используется программная анимация. Максимальный эффект обучения достигается за счет использования трехмерных графических приложений.

В общем понимании трехмерные графические приложения представляют собой специальное программное обеспечение, позволяющее просматривать трехмерные визуальные образы объектов реального мира. При этом при отображении могут преследоваться следующие цели:

- отображение внешнего вида объектов;
- отображение состояния реальных объектов

в виде трехмерных образов.

В первом случае графическое приложение довольно простое, так как создается статическое изображение объекта. Объекты имеют фиксированное состояние (внешний вид) и расположение. Оператор не имеет возможности управлять отображением, изменять или просматривать состояние объекта, получать дополнительную информацию по объектам. Единственное, что может делать оператор, это изменять точку обзора (положение камеры). В простейшем случае приложение может быть сведено к некоторой последовательности изображений (видеофильму). Интерактивность достигается за счет управления воспроизведением при помощи клавиатуры или мыши. Для создания подобных приложений обычно используются специализированные редакторы (редакторы сцен), позволяющие импортировать данные из трехмерных моделей и размещать их на сцене. Написания специального программного кода не требуется. Процесс создания таких приложений состоит из следующих этапов:

1. Разработка моделей в графическом редакторе.
2. Экспорт моделей в редактор сцены, размещение моделей на сцене.
3. Экспорт сцены в проигрыватель.

Если проигрыватель поддерживает непосредственный импорт моделей, созданных в графическом редакторе, то этап создания сцены можно пропустить.

Приложения, отображающие состояния реальных объектов в виде трехмерных образов, имеют более сложную структуру. Помимо отображения трехмерных моделей они позволяют отображать состояние отдельных объектов сцены и их изменения. Разработка таких приложений практически невозможна без написания специального программного кода, так как необходимо задавать связи объектов с некоторыми структурами данных, хранящими их состояние. Данные о состоянии могут передаваться в приложение из базы данных, файла или по сетевому протоколу. Возможен вариант использования некоторого графического редактора (аналог редактора сцен), который позволит не только формировать сцену, но и задавать связи объектов с данными.

В общем случае разработка трехмерных графических приложений под QNX состоит из следующих этапов:

1. Определение требований к графическому приложению. На данном этапе определяется необходимый функционал, который должно реализовывать приложение, требования по производительности, поддерживаемым операционным системам и т. д.

2. Сбор исходных данных для разработки приложений. На данном этапе определяется состав оборудования, отображаемого в графическом приложении, подготавливаются чертежи, схемы, фотографии, технические описания, которые позволят разработать трехмерные модели оборудования, отображаемого в приложении. При этом, так как процесс моделирования автоматизирован, для ускорения процесса разработки чертежи могут быть представлены в цифровом виде.

3. Разработка трехмерных моделей оборудования. На данном этапе в специализированных программах трехмерного моделирования на основе исходных данных, полученных на предыдущем этапе, создаются визуальные образы оборудования. Для данных образов создаются текстуры, настраиваются материалы и т. д. На выходе получаются модели в формате среды разработки.

4. Выбор графического «движка», на основе которого будет разрабатываться приложение.

5. Экспорт модели в формат, воспринимаемый графическим приложением. Данный этап зависит от выбранного графического движка, наличия дополнительных требований и моделей. Этап может дополнительно включать настройку модели, добавления специфических объектов, таких, как камеры, точки обзора и входа в модель, освещение и т. д.

6. Реализация необходимого функционала в соответствии с требованиями к приложению в среде разработки.

7. Загрузка моделей, полученных на этапе 3, в разработанное приложение, настройка модели, определение точек входа в модель и т. д. На выходе данного этапа получается исполняемый файл и набор ресурсов, представляющий собой конечное приложение.

Наиболее популярными форматами описания трехмерных моделей, которые поддерживаются современными графическими движками, являются OBJ, 3DS, BSP.

Формат файлов OBJ – это простой формат данных, который содержит только 3D-геометрию, а именно позицию каждой вершины, связь координат текстуры с вершиной, нормаль для каждой вершины, а также параметры, которые создают полигоны. Формат OBJ интересен в первую очередь тем, что он прекрасно масштабируется и без искажений в него переносится абсолютно любая модель. При экспорте модели в формат .obj на выходе генерируются 2 файла:

1) file.obj, в котором описаны координаты вершин и ссылка на файл с материалами;

2) file.mtl – material texture library, в котором описываются материалы. Материалом условно называется блок этого файла с описанием координат текстур и ссылками на файлы текстур.

Формат файлов 3DS – это формат, содержащий в себе всю информацию о сцене (данные о геометрии, материалах, текстурных координатах, камерах, анимации и т. д.) в бинарном виде. Данные в файле представляются в XML-формате и не могут быть отредактированы вручную. Формат является основным используемым в продуктах Autodesk 3DS Max и фактически является основным форматом для обмена 3D-изображениями.

BSP является форматом файлов для игр серии Quake и игр, основанных на движке id Tech, таких, как Half-Life и Portal. BSP-файлы используют BSP-деревья (алгоритм двоичного разделения пространства) для упрощения сортировки при рендеринге и уменьшения количества полигонов, одновременно выводимых на экран при каждой перерисовке. В силу ограничений, присущих используемому алгоритму, формат не поддерживает работу с открытыми пространствами и сценами, не имеющими ограничений. Для создания сцен в формате BSP существуют специализированные редакторы, поддерживающие импорт моделей из форматов OBJ, 3DS.

Из требований к форматам файлов представления графических моделей вытекают требования к средствам разработки трехмерных моделей. Рассматриваемые средства должны иметь либо непосредственную возможность экспортировать модель в форматы OBJ, 3DS, MESH и BSP, либо иметь расширение (плагин), позволяющее это сделать. На сегодняшний день наиболее распространенными программами трехмерного моделирования, имеющими описанные выше возможности, являются Autodesk 3ds Max и Blender.

Autodesk 3ds Max (панель 3D Studio MAX) – полнофункциональная профессиональная программная система для создания и редактирования трехмерной графики и анимации, разработанная компанией Autodesk. Содержит самые современные средства для художников и специалистов в области мультимедиа. Работает в операционных системах Microsoft Windows и Windows NT (как в 32-битных, так и в 64-битных). Весной 2012 г. выпущена 15-я версия этого продукта под названием «Autodesk 3ds Max 2013».

Blender* – свободный пакет для создания трехмерной компьютерной графики, включающий в себя средства моделирования, анимации, рендеринга, постобработки видео, а также создания интерактивных игр.

Для создания качественной трехмерной модели важно иметь как можно больше информации о создаваемом объекте, поэтому поиск чертежей, схем и фотографий стоит на первом месте. Чертежи необходимы для создания правильной формы объекта и его пропорций, для более подробной детализации модели используются фотографии. Формат исходных данных не имеет принципиального значения, это могут быть как растровые так и векторные форматы, например tga, jpg, png, dwg. После того как материалы найдены, их необходимо импортировать в программу трехмерного моделирования. Затем реализуется процесс моделирования. Важно отметить, что полигонаж модели может быть разным и зависит от поставленных задач, для фотореалистичной визуализации требуется высокополигональная модель, а для менее мощных (например, игровых) движков количество полигонов должно быть минимально. Все модели, используемые в сценах, текстурируются. Это позволяет реалистично имитировать внешний вид поверхностей и отобразить то, что нецелесообразно

* Официальный сайт продукта Blender. [Электронный ресурс.] URL: <http://blender.org>.

но делать на этапе моделирования (например, расшивки, надписи и прочие детали, не обладающие объемом либо он незначителен).

Все текстуры можно разделить на две группы:

- текстуры-плитки;
- текстуры-развертки.

Текстура-плитка – многократно повторяющееся изображение, при стыковке не образует видимых швов и служит для покрытия большой однообразной поверхности без ущерба качеству, но с экономией ресурсов системы.

Текстура-развертка рисуется точно под целевую модель с учетом ее формы и пропорций. На такой текстуре изображаются все характерные детали.

Текстуры можно как рисовать с нуля, так и создавать на основе фотографий. Первый вариант требует некоторых художественных навыков. Наиболее популярным редактором для создания и редактирования текстур является Adobe Photoshop.

Любое графическое приложение в среде Linux\Unix работает на нижнем уровне с использованием реализаций функций OpenGL. OpenGL – это спецификация, определяющая независимый от языка программирования платформеннонезависимый программный интерфейс для написания приложений, использующих двухмерную и трехмерную компьютерную графику. Спецификация включает более 250 функций для рисования сложных трехмерных сцен из простых примитивов. При разработке собственного графического приложения использовать напрямую функции OpenGL нецелесообразно, так как они предназначены только для визуализации и не содержат утилит, позволяющих оптимизировать работу приложения, осуществлять загрузку моделей из файлов, не содержат средств навигации и т. д. При разработке графических приложений используются так называемые графические движки, представляющие собой промежуточное программное обеспечение, главной задачей которого является визуализация трехмерной графики. На сегодняшний день существует множество графических движков, обладающих разным функционалом, производительностью, качеством документации, инструментальных средств. Выбор движка осуществляется на основе требований к графическому приложению, сформулированных на первом этапе разработки. Также от выбранного движка будет зависеть технология разработки трехмерных моделей и экспорта в специфические форматы движка. Этап выбора графического движка является наиболее сложным и трудоемким из вы-

шеперечисленных, так как помимо выбора собственно самого движка он включает портирование движка на целевую операционную систему QNX.

При выборе графического движка следует руководствоваться следующим критериями:

1. Наличие открытого кода. В силу того, что под QNX в открытом доступе не существует собранных графических движков, обычно при выборе движка идет речь о портировании движка с другой платформы. Портирование подразумевает компиляцию исходных кодов под операционную систему (ОС) с учетом специфики платформы: наличия библиотек, примитивов и системных функций ОС.

2. Поддержка работы в ОС семейства UNIX (в идеале работа в QNX). Если в движке не заявлена поддержка Unix-систем, то вероятность портировать его под QNX стремится к нулю.

3. Тип рендеринга, поддержка аппаратного рендеринга (аппаратного ускорения). Существуют движки, поддерживающие только программный рендеринг, а также движки, поддерживающие и аппаратный.

4. Производительность движка, измеряемая в FPS, определяет максимальное количество кадров в секунду.

5. Поддержка функций визуализации только содержимого, попадающего в поле зрения (без визуализации частей моделей за кадром/не видимых на кадре).

6. Возможность работы со сценами: объединение объектов в сцены, динамическая загрузка сцен с целью организации перехода между объектами (к примеру, вход в помещение, переход к более детальному отображению и т. д.).

7. Наличие встроенных функций скрытия/отображения отдельных объектов.

8. Наличие встроенных средств навигации (изменения положения камеры).

9. Наличие контроллеров, отслеживающих нажатия на объекты, выбор манипулятором «мышь» и т. д.

10. Поддержка часто используемых форматов трехмерных моделей типа 3DS, OBJ без дополнительной конвертации.

11. Наличие и качество документации.

Наиболее популярными, с точки зрения указанных критериев, являются следующие графические движки:

id Tech 3;
Irrlicht;
OGRE.

• id Tech 3 – игровой движок, разработанный компанией id Software специально для игры Quake III Arena и с успехом используемый в других проектах. Для данного движка опубликованы исходные коды по лицензии GNU GPL 2.0, последняя версия 1.32 выпущена 19 августа 2005 г. Движок поддерживает ОС Windows, GNU/Linux, Mac OS. Исходный код портированного движка выложен на официальном сайте QNX и доступен для скачивания. Движок написан на языке программирования C, не использует дополнительных прослоек, библиотек и т. д. После сборки портированного движка получается непосредственно игра Quake 3 со своими заставками, меню и т. д. Отсутствие комментариев в исходном коде и документации значительно усложняют доработку движка, для которого не известны ни архитектура, ни потенциальные возможности. Отсутствие объектно-ориентированной основы не позволяет проследить связи между компонентами движка. Вместе с тем движок может быть без дополнительной доработки адаптирован для задач отображения, в которых не требуется дополнительный функционал. В редакторе GtkRadiant создается сцена, которая загружается в движок. Оператор может перемещаться по сцене, смотреть внешний вид оборудования.

• Irrlicht (Irrlicht Engine*) – трехмерный графический движок, который является бесплатным свободным программным продуктом и распространяется на условиях лицензии zlib. Irrlicht использует возможности OpenGL и нескольких собственных рендереров. Пользователю предоставляются различные функциональные возможности по загрузке и управлению трехмерными (3D) объектами (сцены, модели и т. п.), немногими спецэффектами и графическим интерфейсом пользователя. Не требует подключения сторонних модулей для реализации высокоуровневых функций. Существует 3 официальных дополнения для Irrlicht: IrrKlang (аудиобиблиотека), IrrXML (загрузка и обработка XML-файлов), IrrEdit (редактор сцен). Для использования расширенных функций моделирования физических явлений дополнительно применяется физический движок ChronoEngine. Одна из важных особенностей Irrlicht – его кроссплатформенность, т. е. способность работать на различных платформах. Платформенезависимая прослойка обеспечивает легкую портируемость

на различные не поддерживаемые официально платформы, в частности существуют порты под android, iPhone и пр. Последняя версия 1.7.3 (23 февраля 2012 г.). Данный движок основывается на библиотеке SDL (Simple DirectMedia Layer) SDL – это свободная кроссплатформенная мультимедийная библиотека, реализующая единый программный интерфейс к графической подсистеме, звуковым устройствам и средствам ввода для широкого спектра платформ. Данная библиотека активно используется при написании кроссплатформенных мультимедийных программ*. SDL поддерживает большое количество ОС, в том числе QNX. Irrlicht является полностью объектно-ориентированным, имеет большой объем документации, прост в использовании. С аппаратными средствами работает через прослойку SDL, что сказывается на снижении производительности при визуализации трехмерных образов. Также полигоны модели не хранятся в видеопамяти, а каждый раз при визуализации загружаются в нее по шине из оперативной памяти, поэтому движок замедляет воспроизведение сильно нагруженных сцен вне зависимости от параметров видеокарты. Сгладить этот недостаток можно, например, определяя вручную, какие объекты видны данной камере, а остальные объекты делать «невидимыми». Главным недостатком данного движка является требование запуска X-сервера (графической системы, портированной с ОС Linux) для поддержки аппаратного ускорения. При этом запуск X-сервера под QNX при использовании встроенных графических чипсетов Intel невозможен по причине отсутствия реализации совместимого OpenGL agpgart-драйвера. Этот факт свидетельствует о невозможности использования данного движка с аппаратным ускорением.

• OGRE – объектно-ориентированный графический движок с открытым исходным кодом, написанный на C++. OGRE изначально создан как графический движок для рендеринга трехмерной графики. Большую популярность движок получил за счет своей гибкости, что позволяет «скрещивать» его со многими другими библиотеками (физика – ODE, Newton, PhysX, Bullet; звук, сеть, графический интерфейс и т. д.). С помощью усилий многих профессионалов появились библиотеки, адаптированные для работы с OGRE. Например, для реализации физики в приложении

* Официальный сайт продукта Irrlicht. [Электронный ресурс.] URL: <http://irrlicht.sourceforge.net/>.

* Официальный сайт продукта Simple Directmedia Layer. [Электронный ресурс.] URL: <http://www.libsdl.org/>.

ях, использующих данный движок, портированы такие библиотеки, как PhysX SDK (NxOgre), Newton Game Dynamics (OgreNewt), Bullet Physics Library (OgreBullet), Open Dynamics Engine (Ogre-ODE). Данный движок зависит от библиотек FreeType2 (<http://www.freetype.org/>), FreeImage (<https://github.com/lynxluna/freeimage>), zlib (<http://zliblib.sourceforge.net/>), OpenIL (<http://openil.sourceforge.net/>), которые также должны быть портированы перед портированием движка. OGRE не имеет своей реализации обработки событий клавиатуры и манипулятора «мышь», основываясь на OIS (Object Oriented Input System). Поэтому перехват системных событий требуется реализовать на уровне движка.

В качестве среды разработки приложений для QNX обычно используется IDE Momentics*. В данной среде разработки реализуются этапы кодирования, верификации и сопровождения программного кода. Комплект поставляется в виде трех дистрибутивов, предназначенных для функционирования, соответственно, в среде операционных систем Windows, Linux и QNX Neutrino. Выбор операционной системы для установки QNX Momentics зависит от используемых подходов к разработке и личных предпочтений программистов. QNX Momentics включает в себя:

1. Инструменты для разработки и отладки программ для всех аппаратных платформ, поддерживаемых OCPB QNX Neutrino (ARM, MIPS, PowerPC, SH-4, x86).

2. Инструменты для оптимизации программ с помощью прикладного и системного профилирования, анализа памяти и покрытия кода.

3. Инструменты для построения целевых систем QNX Neutrino.

4. Инструменты визуального мониторинга целевых систем QNX Neutrino во время их функционирования.

5. Инструмент для визуального построения графических интерфейсов пользователя – Photon Application Builder.

6. Средства удаленного доступа к рабочему столу Photon microGUI целевой системы QNX Neutrino из сред Windows и UNIX.

Существуют различия при запуске 3D-приложений, разработанных в IDE Momentics в консольном режиме и в графической оболочке Photon. При работе в консольном режиме удаленный запуск приложения может осуществлять только

один разработчик, так как консоль имеет один графический контекст, который инициализируется монополично одним процессом. Таким образом, отладка консольных приложений, использующих 3D-графику, должна производиться разработчиками поочередно или на различных стендах. Запуск и отладка 3D-приложений в графической системе Photon свободны от ограничений консольного запуска, так как графический контекст инициализируется независимо для каждого окна. Поэтому пользоваться одним стендом для отладки могут несколько разработчиков одновременно.

Трехмерные графические приложения ресурсоемки. В связи с этим актуальным является вопрос оптимизации работы графического приложения, который сводится фактически к решению трех задач:

- оптимизация исходной модели;
- оптимизация отображения моделей;
- оптимизация кода приложения.

Оптимизация исходной модели обычно сводится к уменьшению количества полигонов и уменьшению количества объектов. Уменьшение количества полигонов может достигаться снижением детализации малозначимых элементов сцены, вплоть до перевода их на уровень текстур (если объект не значим, то он может быть просто нарисован на текстуре).

Помимо количества полигонов, например для движка OGRE, большое значение имеет количество отображаемых объектов. При подготовке моделей для движка OGRE следует использовать формат mesh. Файл mesh содержит описание одного объекта сцены (корпуса, вентиля, манометра и т. д.). Чем больше отдельных объектов, тем больше файлов с данными, тем медленнее будет рендериться сцена, поэтому рекомендуется при разработке моделей сокращать количество объектов, сокращая таким образом количество mesh-файлов. Сокращение количества объектов может достигаться объединением нескольких объектов в один (к примеру, все переборки корабля, загородки, элемент на палубе могут быть объединены в один объект (mesh), который будет называться «корпус корабля»).

Оптимизация отображения моделей в самом простом случае сводится к снижению детализации отображаемых объектов в зависимости от расстояния до них. Для этих целей обычно используется технология LOD. LOD* (англ. Levels

* Официальный сайт компании «СВД Встраиваемые Системы». [Электронный ресурс.] URL: <http://www.kpda.ru>.

* Свободная энциклопедия. [Электронный ресурс.] URL: <http://ru.wikipedia.org/>.

Of Detail – уровни детализации) – прием в программировании трехмерной графики, заключающийся в создании нескольких вариантов одного объекта с различными степенями детализации, которые переключаются в зависимости от расстояния объекта до виртуальной камеры. Смысл приема заключается в том, чтобы не отображать высокодетализированные объекты, находящиеся на большом расстоянии. Другой метод заключается в использовании одной основной, «грубо приближенной», модели и нескольких внешних надстроек к ней. Каждая последующая надстройка к основной модели дополняется элементами детализации пропорционально номеру надстройки, т. е. на самом большом расстоянии будет отображаться единственная главная модель объекта. С приближением же последнего к камере, к конвейеру отрисовки будут последовательно подключаться последующие надстройки деталей. Использование LOD способно существенно снизить требования к ресурсам компьютера при выводе графики на экран. Но в любом случае для использования технологии LOD необходимо разработать несколько вариантов трехмерных моделей. Простейшим вариантом использования технологии LOD, не требующим разработки дополнительных моделей, является простое скрытие удаленных объектов.

Другой способ оптимизации отображения модели заключается в создании единственной текстуры среднего размера с достаточной детализацией и наложении ее на полигоны объекта во всех случаях, когда требуется визуализация. Однако здесь возникают две проблемы. Во-первых, если текстура создана с максимально возможным разрешением и детализацией (например, 2048×2048 пикселей при глубине цветового охвата 32 бит), то расход вычислительных ресурсов и памяти при наложении текстур на все видимые объекты в трехмерной сцене будет слишком большим. Во-вторых, если уменьшить размер (и тем самым детализацию) текстур, то объекты по мере приближения к плоскости проекции будут выглядеть все более грубо. Для решения указанных проблем разработана технология, получившая название MIP mapping. Ее суть заключается в предварительном или динамическом создании набора текстур с различным разрешением и уровнем детализации на основе базовой текстуры максимального разрешения. При построении трехмерной сцены определяется удаленность полигона от картинной плоскости и соответственно этому

значению накладывается текстура с заданным разрешением. Важно отметить, что, например, графический движок OGRE имеет встроенную возможность автоматически использовать инструменты MIP-mapping.

Современные графические движки позволяют подключать плагины, которые также оптимизируют отображение сложных моделей. Например, графический движок OGRE имеет стандартный плагин под названием Plugin_PCZSceneManager (Portal Connected Zone Scene Manager). Он оставляет особый вид менеджера сцены. Его основная идея заключается в том, что вся сцена разбита на зоны (кубы или шары) и все объекты принадлежат хотя бы одной из зон. Соседние зоны могут быть соединены между собой так называемыми порталами (с геометрической точки зрения – часть поверхности зоны). Порталы всегда существуют парами (если у зоны А есть портал в зону Б, то у зоны Б обязательно есть портал в зону А, имеющий то же расположение). Также зоны могут содержать антипорталы (некие объекты, ограничивающие видимость, которые сами по себе никак не отображаются). Суть в том, чтобы не рендерить невидимые объекты. А видимы они или нет, определяется следующим образом: начиная с зоны, в которой находится камера, движок начинает рендерить все объекты и добавлять в список текущих «перегородок, ограничивающих обзор», антипорталы этой зоны. Далее перебираются все порталы этой зоны и если они не закрыты антипорталами – рекурсивно переходят в зону, куда вел данный портал. Обычно имеется зона по умолчанию (присутствующая всегда), которая содержит все объекты, не привязанные к какой-либо еще зоне. Просто разбить сцену на зоны и соединить их порталами не имеет смысла, пока не добавлены антипорталы, так как, пока их нет, в процессе вывода на экран будут перебираться все зоны. Если сделать зону, в которую не ведет ни один портал, то никакие объекты из нее не будут рендериться (даже если не установлено ни одного антипортала). Так будет, если текущая камера находится вне этой зоны, в противном случае будут видны объекты данной зоны и только они.

Основной сложностью использования данной техники является выбор принципа выделения зон и построения порталов. С одной стороны, объекты могут привязываться к зонам при разработке приложения на языке программирования, что достаточно трудозатратно. Возможен другой под-

ход: в процессе подготовки модели логические объекты, находящиеся за дверьми или перегородками (например, каюты внутри корабля), выделяются прозрачными кубами, имеющими имена с заданным префиксом. Аналогично предлагается поступать с переходами (порталами) – называть двери и перегородки согласованными именами. В таком случае на уровне движка возможно единообразно выделить набор зон и порталов, тем самым обеспечив универсальность при построении моделей и отсутствие задания моделей в коде 3D-приложения.

Оптимизация работы графического приложения может осуществляться тремя способами:

- полностью программная реализация функций OpenGL, при этом все расчеты выполняются центральным процессором с использованием оперативной памяти компьютера;

- частичное выполнение функций графическим микроконтроллером, при этом используется оперативная память компьютера и часть расчетных операций выполняется центральным процессором;

- выполнение всех функций OpenGL на графическом процессоре, при этом в полной мере используется память видеокарты, в нее загружается вся модель и текстуры. Достигается наибольшая производительность, но возрастают требования к объему памяти видеокарты. При этом уменьшаются требования к центральному процессору, так как в случае графических приложений он выполняет обслуживающие функции.

Таким образом, наибольшая производительность работы графических приложений достигается при наличии аппаратного ускорения. Под аппаратным ускорением [см. лит.] понимают применение аппаратного обеспечения для выполнения некоторых функций быстрее по сравнению с выполнением программ процессором общего назначения. При работе с графикой в качестве аппаратного обеспечения применяются графические процессоры (или видеокарты). С аппаратным 3D-ускорением трехмерные изображения создаются графическим процессором видеокарты, при этом не используются ресурсы центрального процессора. Без 3D-ускорения процессор вынужден отрисовывать все самостоятельно, используя библиотеки, которые требуют значительной вычислительной мощности. Основное отличие аппаратного ускорения от программного за-

ключается в параллельности, позволяя аппаратному обеспечению быть гораздо быстрее, чем программное. Аппаратные ускорители специально спроектированы для программного кода, создающего высокую вычислительную нагрузку. В зависимости от степени детализации аппаратное ускорение может варьироваться от небольшой функциональной единицы до крупного функционального блока, как, например, обработка 3D-изображения. Работа с трехмерными графическими изображениями довольно специфична с точки зрения вычислений, поэтому специализированные графические процессоры справляются с этой задачей лучше, чем центральный процессор, предназначенный для решения широкого круга задач. Стандартная поставка QNX предназначена для архитектуры x86 и включает в себя все необходимые драйверы для поддерживаемых аппаратных средств. Для других архитектур встраиваемых систем существуют специальные сборки и так называемые BSP (Board Support Package) пакеты [см. лит.] – интегрированные пакеты драйверов и/или модулей операционной системы, реализующих поддержку определенной аппаратной платформы. По сути BSP является модулем, набором модулей или набором драйверов устройств, встраиваемым в операционную систему и реализующим поддержку всего оборудования и особенностей конкретной аппаратной платформы.

В заключение необходимо отметить, что создание КОС с использованием трехмерных графических отображений дает максимальный эффект обучения и наглядности представления учебного материала, однако требует в процессе разработки внимания ко многим технологическим аспектам. Описанные в статье особенности создания графических приложений в среде QNX необходимо учитывать на всех этапах разработки: формулирование требований к приложению, проектирование пользовательского интерфейса и архитектуры приложения, выбор аппаратной платформы, создание трехмерных моделей, разработка исходного кода приложения, сборка приложения и настройка среды выполнения.

СПИСОК ЛИТЕРАТУРЫ

Кртен Р. Введение в QNX Neutrino. СПб.: БХВ-Петербург, 2011.

E. L. Kalishenko, Yu. V. Romanenko

TECHNOLOGICAL ASPECTS OF REALIZATION OF COMPUTER TRAINING SYSTEMS WITH THREE-DIMENSIONAL GRAPHIC IN QNX ENVIRONMENT

Article covers technological aspects of 3D-applications development as part of computer-based training systems. Graphical applications optimization techniques are considered for QNX realtime operating system.

Computer-based training, simulators, 3D-applications, QNX operating system, graphical applications optimization

УДК 004.41

С. А. Романенко, А. С. Скрипникова

Управление процессом разработки на основании требований

Рассматривается процесс разработки программных изделий, применяемый в открытом акционерном обществе «Научно-инженерный центр Санкт-Петербургского электротехнического университета». Обсуждаются особенности подхода, его достоинства и недостатки, сложности, возникавшие в процессе его внедрения и применения. Подчеркивается положительное влияние выбранного подхода на процесс взаимодействия заинтересованных лиц на всех этапах жизненного цикла разрабатываемых программных изделий.

Домен, матрица трассировки, оптимизация процессов, предметно-ориентированное проектирование, процесс разработки, управление требованиями

С начала апреля 2011 г. в открытом акционерном обществе «Научно-инженерный центр Санкт-Петербургского электротехнического университета» (ОАО «НИЦ СПбЭТУ») ведутся работы по внедрению на уровне предприятия процесса разработки программных изделий на основании требований. Целью внедрения описываемого процесса является оптимизация разработки за счет снижения временных затрат на управление, издержек, связанных с доработкой программных изделий, а также достижения требуемого уровня качества выпускаемых программных изделий.

На сегодняшний день описываемый процесс внедрен в пилотной зоне, в качестве которой выбрана группа проектов по разработке программных продуктов для Федеральной таможенной службы Российской Федерации (ФТС России). В данной статье рассматриваются результаты поэтапного внедрения, приводятся решения возникших проблем.

На момент принятия решения о необходимости внедрения существующий процесс разработки программных изделий не содержал в себе от-

дельно выделенных подпроцессов сбора, описания требований и управления ими. Поэтому возникла необходимость их выделения и формализации.

В соответствии с международным стандартом IEEE Std 610.12 требованием было принято считать условие или возможность, требуемые пользователем для решения задач или достижения цели; условие или возможность, которые должны удовлетворяться системой (компонентом системы), или которыми система (компонент системы) должна обладать для обеспечения условий контракта, стандартов, спецификаций или других регулирующих документов.

Учитывая требования международного стандарта IEEE Std 830-1998 и особенности существующего на тот момент процесса разработки, в качестве атрибутов требования были выделены:

1. Идентификатор требования (обязательный).
2. Наименование требования (обязательный).
3. Тип требования (обязательный).
4. Описание требования (обязательный).
- 4.1. Перечень потребителей (обязательный).

4.2. Состав и структура входных и выходных данных, ограничения (условия) на их значения (обязательный).

4.3. Описание или формализованные модели алгоритмов реализации требования (обязательный для следующих типов требований: «функциональные требования», «ограничения»).

4.4. Математическое обоснование (необязательный).

4.5. Описание прототипов интерфейса пользователя (если интерфейс пользователя необходим для реализации требования) (обязательный для следующих типов требований: «внешние интерфейсы»).

4.6. Спецификации и описание протоколов взаимодействия (необязательный).

4.7. Ограничения на реализацию требования (необязательный).

5. Наименования компонентов (обязательный).

6. Способ проверки выполнения требования (обязательный).

7. Описание взаимосвязи с другими требованиями (обязательный).

8. Источник требования (обязательный).

9. Приоритет (обязательный).

10. Срок реализации (обязательный).

На основании этих данных был сформирован стандарт предприятия «Постановка задачи. Требования к содержанию и оформлению» (далее Стандарт), который ввел понятие документа «Постановка задачи» и формализовал процесс его создания. Применение данного Стандарта на практике позволило еще на стадии описания проверить полноту и непротиворечивость требований, используя атрибут «Способ проверки выполнения требования», и отследить изменения во взаимосвязанных требованиях за счет наличия типизированных связей. Более того, получаемое описание стало основой для создания тестовых сценариев проверки реализуемого функционала, сократив тем самым время на подготовку тестового окружения, а значит, и время на подготовку версии выпускаемого программного изделия. Однако временные затраты на создание документов с описанием постановки задачи в значительной мере возросли за счет необходимости ручного отслеживания взаимосвязей и ввода однотипных данных. В связи с этим было принято решение об автоматизации подпроцесса описания требований.

Для этих целей было выбрано программное средство для управления бизнес-процессами JIRA. В соответствии со Стандартом в данном программном средстве были настроены формы для описания требований. Это позволило сократить временные затраты за счет использования разного рода справочной информации и автоматизированной работы со связями. Практика применения JIRA показала, что при таком варианте реализации все участники процесса разработки программного изделия в любой момент времени имеют доступ к любому из требований и в автоматическом режиме получают информацию о имеющихся между ними связях, что в значительной мере упростило коммуникации внутри проектной команды.

Параллельно с формализацией подпроцесса описания требований большое внимание уделялось также подпроцессу сбора требований. На момент принятия решения о необходимости внедрения рассматриваемого процесса за сбор требований практически полностью отвечал аналитик проекта. Он выступал в качестве посредника между конечными пользователями и заказчиком (с одной стороны) и проектной командой, полностью взяв на себя обязанности по преобразованию пользовательских требований и требований, содержащихся в техническом задании на разработку, в формализованное описание требований, соответствующее принятому Стандарту. С одной стороны, это давало положительный эффект от того, что все коммуникации между заинтересованными лицами шли через одного человека, что не приводило к искажению информации при передаче ее между несколькими посредниками. С другой стороны, аналитику приходилось постоянно общаться на «двух языках»: на языке предметной области заказчика и пользователей и языке технических специалистов проектной команды. Опыт разработки программных продуктов с использованием такого взаимодействия показал, что в значительной мере возрастают риски, связанные с работой аналитика, его постоянной доступностью для заинтересованных лиц, способностью быстро и точно транслировать запросы заказчика и пользователей проектной команде и наоборот. Для того чтобы снизить указанные риски, было принято решение обратиться к предметно-ориентированному проектированию (domain-driven design), суть которого заключается в создании исключительно простых (при этом не примитивных) моделей, понятных всем участникам

процесса разработки. Изучив особенности данного подхода [1] и его внедрения [2], было принято решение о поэтапном внедрении в процесс управления разработкой программных изделий. На сегодняшний день для выбранной в качестве пилотной зоны группы проектов введено понятие домена и универсального языка.

Для понимания и описания ключевых деталей, специфичных для предметной области, абсолютно необходимо использование непротиворечивого и четкого языка. Под языком в предметно-ориентированном проектировании понимаются концепции и их связи. Универсальный язык одинаково понятен всем участникам процесса. Использование универсального языка внутри проектной команды позволило ее участникам более глубоко вникнуть в предметную область, понять потребности заказчика и пользователей. Кроме того снизились риски, связанные с работой аналитика, за счет отсутствия необходимости преобразования терминологии, используемой заказчиком и пользователями, в терминологию, используемую проектной командой.

Следующим шагом внедрения предметно-ориентированного проектирования в подпроцесс сбора требований стал ввод понятия «Домен», который формируется в ходе стратегического проектирования. Домен представляет собой совокупность наиболее общей и целостной модели и универсального языка. В центре стратегического проектирования и получаемого в результате домена лежит то множество частей, из которых состоит общая модель, а также взаимосвязи между этими частями. Такие части, которые при предметно-ориентированном проектировании можно считать «малыми моделями», существуют в ограниченных контекстах. Для каждой «малой модели» определяется контекст, в котором она существует. При работе с контекстами создавались максимально выразительные модели – такие, которые дают представление именно о назначении системы, а не о конкретной реализации. Когда удастся этого достичь, то концепции предметной области, определяющие универсальный язык, естественным образом оказываются «на поверхности». Модели получают гибкими и удобными для дальнейших изменений. В результате проектирование способа реализации, описываемое в соответствии со Стандартом в виде формализованных требований, отходит на второй план и происходит всегда к контексте спроектированного домена.

Опыт применения внедренных подходов предметно-ориентированного проектирования показал, что проектирование домена до момента описания требований и проектирования способов реализации позволяет в значительной мере снизить временные затраты на разработку программного изделия. Снижение достигается за счет уменьшения количества ошибочных решений, связанных с недостаточным пониманием целей и потребностей заказчика и пользователей, которые приводят к необходимости повторного проектирования и реализации части функционала. Кроме того в значительной степени упрощается коммуникация и увеличивается вовлеченность заказчика и пользователей в процесс разработки программного изделия, так как модели, построенные в рамках проектирования домена, просты и понятны всем участникам в отличие от существовавших ранее технических моделей, описывающих исключительно варианты реализации. Основной сложностью при использовании подходов предметно-ориентированного проектирования является необходимость выделения значительного времени на проектирование домена и разработку универсального языка. При этом должны пресекаться любые попытки проектирования реализации. После получения некоторого опыта и вовлеченности всех участников проектной команды в процесс проектирования этот этап естественным образом вошел в процесс разработки программных изделий.

Следующим шагом внедрения процесса управления разработкой программных изделий на основании требований стало введение подпроцесса управления требованиями, включающего идентификацию, сбор, описание, анализ, отслеживание, приоритезацию требований, достижение соглашения по требованиям и затем управление изменениями на всем жизненном цикле разрабатываемого изделия. В соответствии с рекомендациями стандарта ANSI PMI PMBOK® A Guide to the Project Management Body of Knowledge (PMBOK® Guide) [3] для этих целей была введена матрица взаимосвязи или трассировки требований, содержащая информацию об источнике требования, связанных с данным источником формализованных описаниях, подготовленных в соответствии со Стандартом, приоритетах, сроках разработки, тестовых сценариях и разделах документации, содержащих описание реализованного функционала. Использование матрицы трассировки позволило связать между собой ре-

зультаты таких подпроцессов, как сбор и описание требований, реализацию требований, тестирование, документирование результатов. Применение матрицы трассировки в процессе управления разработкой программных изделий позволило достичь ряда положительных результатов. Снизилась временные затраты на контроль полноты выполнения требований технического задания за счет того, что матрица трассировки позволяет увидеть полную картину покрытия высокоуровневых требований, описанных в техническом задании, детальными и формализованными требованиями, описанными в соответствии с принятым Стандартом. Выпускаемые версии программных продуктов более полно соответствуют текущим потребностям заказчика и пользователей, что было достигнуто за счет разработки в первую очередь более приоритетных требований и возможности изменения приоритетов в зависимости от потребностей. Достигнуто снижение трудоемкости разработки программного продукта за счет отсекаемых требований, которые не являются потребностями заказчика и отсутствуют в техническом задании на разработку. Детализация высокоуровневых требований и управление приоритетами позволили, при сокращении сроков выпуска версии программного изделия, выпускать версию, содержащую минимальный допустимый набор функций, расширяя в дальнейшем, с выпуском следующей версии, наполнение каждой функции. Данный подход позволяет конечным пользователям приступить к работе сразу после выпуска версии, не дожидаясь выхода последующих. Еще одним положительным результатом применения матрицы трассировки стало сокращение временных затрат на тестирование и написание документации для конечных пользователей в случае изменения отдельных функций разрабатываемого программного изделия, что достигается за счет четких связей между требованием на разработку, связанными с ним тестовыми сценариями и разделами документации, описывающими реализацию изменяемой функции. Кроме того использование матрицы трассировки позволило упростить процесс управления ожиданиями пользователей. Каждый запрос конечного пользователя фиксируется в матрице трассировки. Появление в соответствующем разделе матрицы информации по данному запросу позволяет говорить об изменении его статуса. Например, при появлении идентификаторов тестовых сценариев, проверяющих

требование, можно говорить, что версия, содержащая изменения, запрашиваемые пользователем, находится на тестировании. Таким образом, матрица трассировки на данном этапе является основным источником информации для процесса управления разработкой программного изделия, позволяя не только оценивать текущее состояние процесса, но и осуществлять как краткосрочное, так и долгосрочное планирование. Стоит отметить, что при таком количестве положительных результатов нельзя не обратить внимание на тот факт, что опыт построения матриц трассировки для проектов, находящихся на момент начала внедрения описываемого процесса на завершающих стадиях разработки или сопровождении, показал достаточно высокую сложность и трудоемкость построения матриц. В данном случае был применен упрощенный подход. Матрица составлялась не на программное изделие в целом, а лишь на версию, выпускаемую на момент внедрения практики использования матрицы трассировки. Такой подход незначительно увеличил трудоемкость по данным проектам, но при этом позволил создать отправную точку для дальнейшего расширения матрицы трассировки при планировании и разработке следующих версий. С целью сокращения временных затрат на построение матрицы используемое для описания требований программное средство JIRA было настроено таким образом, чтобы на основании сведений, указанных в описании каждого требования, и информации о взаимосвязи требований матрица трассировки строилась в автоматическом режиме. Автоматическое построение обеспечило возможность получения актуальных сведений о состоянии процесса разработки программного изделия в любой момент времени.

Таким образом, если обобщенно описать процесс управления разработкой программных изделий, внедренный в ОАО «НИЦ СПбЭТУ», то он представляет собой создание и управление матрицей трассировки требований в контексте спроектированного домена. Такой процесс показал свою эффективность. Он позволяет в значительной мере снизить временные и трудозатраты на разработку. Более того, позволяет вовлечь заказчика и конечных пользователей в процесс проектирования, что не характерно для проектов с государственным участием, которыми являются проекты пилотной зоны. Это позволило приблизить процесс разработки программных изделий для государственных заказчиков к процессу раз-

работки коммерческих программных изделий, повысить качество выпускаемой продукции. При этом процесс достаточно гибкий, за счет изменения структуры матрицы трассировки можно добиться управления любым количеством важных для проекта атрибутов. В ходе внедрения и использования процесса был выявлен ряд проблем, решения которых были найдены и описаны в данной статье. В будущем планируется внедрение описанного процесса управления разработкой

программных изделий на всем предприятии. На сегодняшний день развитие процесса не прекращается. Успешный опыт использования подходов предметно-ориентированного проектирования позволяет предположить, что его внедрение уже непосредственно в этап проектирования реализации будет способствовать дальнейшему улучшению процесса, уменьшению трудоемкости и временных затрат на разработку.

СПИСОК ЛИТЕРАТУРЫ

1. Эванс Э. Предметно-ориентированное проектирование (DDD): структуризация сложных программных систем / Пер. с англ. М.: Вильямс, 2011.
2. Vaughn V. Implementing Domain-Driven Design. Addison-Wesley, 2013.

3. A Guide to the Project Management Body of Knowledge (PMBOK® Guide). 4th ed. Newtown Square (PA) / Project Management Institute. 2008.

S. A. Romanenko, A. S. Skripnikova

THE DEVELOPMENT PROCESS BASED ON THE REQUIREMENTS

The article deals with the development process of software products. The features of the process, its advantages and disadvantages, the difficulties that arose in the course of its implementation and use, are discussed. The positive impact of the process chosen by the interaction of stakeholders at all stages of the life cycle of software products is underlined.

Domain, traceability matrix, process optimization, domain-driven design, process development, requirements management

УДК 681.5.017

О. С. Черная, Ю. Ю. Федорова, С. А. Беляев

Применение методов и средств автоматизированного тестирования для проверки качества программных комплексов обработки измерительной информации

Описывается подход к построению обобщенных сценариев автоматизированного тестирования, показана структура классов, позволяющая строить такие тесты, рассматриваются преимущества и недостатки указанного подхода.

Автоматизированное тестирование, обобщенные сценарии, WindowTester Pro

Уровень сложности современных программных комплексов настолько велик, что уже никого не удивит переходом от ручного тестирования к автоматизированному. При выполнении ручного тестирования проверка функционирования осуществляется человеком, который вводит данные, изменяет состояние элементов управления и контролирует поведение программного комплекса.

При этом время тестирования существенно зависит от количества элементов ввода-вывода данных и возможностей по использованию элементов управления, сложность растет экспоненциально. Ключевыми недостатками данного подхода являются время проверки, которое зависит от количества проверяемых комбинаций, и «человеческий фактор».

Автоматизированное тестирование использует программные средства для выполнения тестов и проверки результатов, что помогает сократить время тестирования и упростить его процесс*. При этом существует 2 основных подхода: тестирование на уровне кода, в частности, модульное тестирование, и тестирование интерфейса пользователя, когда осуществляется имитация действий пользователя с помощью специальных тестовых фреймворков. Современные инструменты тестирования интерфейса пользователя в общем случае поддерживают не только выполнение сценария теста, но и его запись, когда действия пользователя записываются и затем могут воспроизводиться в автоматическом режиме. Сценарии сохраняются в формате, поддерживаемом конкретным фреймворком, и могут изменяться пользователем перед выполнением. В частности, фреймворк WindowTester Pro** сохраняет сценарии в формате модульных тестов на языке Java, что позволяет объединить в одном инструменте как модульное тестирование, так и тестирование интерфейса пользователя.

Ключевые преимущества автоматизированного тестирования***:

- исключение «человеческого фактора» – автоматическое выполнение сценария не пропустит тест «по неосторожности» и не ошибется в результатах, что повышает качество результата;
- быстрое выполнение – автоматическое тестирование не требует сверки с инструкциями и документацией;
- небольшие затраты на поддержку – когда сценарии уже написаны, на их поддержку и анализ результатов требуется, как правило, меньше времени, чем на проведение того же объема тестирования вручную;
- автоматическое формирование отчетов о результатах тестирования;
- непрерывное выполнение тестов, в том числе и в «нерабочее время».

Ключевые недостатки автоматизированного тестирования****:

- все написанные сценарии всегда будут выполняться однообразно, что не позволит обнару-

жить ошибку, которая была бы «видна» человеку, например, на экранной форме отобразились не те данные, но сценарий предусматривает проверку других элементов;

- чем чаще изменяется приложение, тем выше затраты на поддержку сценариев;

– разработка автоматизированных тестов – это сложный процесс, так как фактически идет разработка приложения, которое тестирует другое приложение, что повышает требования к знаниям и умениям тестировщика.

Таким образом, выбор подхода к тестированию существенно зависит от особенностей тестируемого приложения и процесса его разработки, а также квалификации тестировщика и требований к качеству.

Основные особенности программных комплексов обработки измерительной информации:

- обработка большого массива исходных данных;
- наукоемкие подходы по обработке данных;
- отличия между программными комплексами заключаются в основном в способе обработки данных, а не в способах представления.

Для разработки программных комплексов обработки измерительной информации используется модель инкрементальной разработки, когда разные части программного кода разрабатываются в разное время и в общем случае не должно возникать изменений в уже завершенных частях.

Учитывая указанные особенности программных комплексов и процесса их разработки, появляется возможность применять автоматизированное тестирование. Рассмотрим в качестве языка разработки – Java, в качестве средства тестирования – WindowTester Pro. В результате сценарии генерируются на языке Java в виде Unit-тестов. В связи с незначительными отличиями в интерфейсе пользователя возможна однократная генерация сценария, а после простой доработки Java-кода – получение однотипных сценариев для всех программных комплексов. Такой подход позволяет с использованием рефакторинга [см. лит.] выделить в сценариях общие части и сформировать обобщенный сценарий, который обеспечит тестирование нескольких программных комплексов.

Основные отличия программных комплексов обработки измерительной информации, которые должны учитываться при формировании обобщенных сценариев:

- отличия в интерфейсе пользователя;
- отличия в алгоритмах, реализующих серверную логику;
- отличия в исходных данных;

* Автоматизированное тестирование // <http://ru.wikipedia.org/wiki/Автоматизированное%20тестирование>.

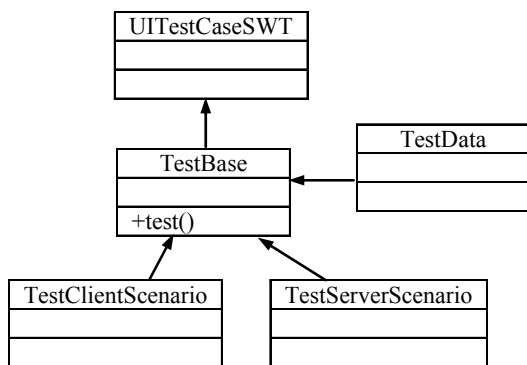
** WindowTester Pro User Guide // <https://developers.google.com/java-dev-tools/wintester/html/index>.

*** Автоматизация тестирования программных систем // <http://habrahabr.ru/post/160257/>.

**** Там же.

– особенности процесса работы конкретного программного комплекса.

С учетом перечисленных особенностей разработана структура классов, позволяющая создавать обобщенные сценарии (рисунок). `UITestCaseSWT` – базовый класс, обеспечивающий тестирование. В рамках `TestBase` реализуется базовая функциональность, позволяющая исполнять обобщенные сценарии в интересах всех программных комплексов обработки измерительной информации, `TestData` предоставляет уникальные данные, `TestClientScenario` позволяет тестировать уникальные особенности клиентских приложений, `TestServerScenario` – серверных приложений.



Использование обобщенных сценариев изменяет классическую технологию автоматизированного тестирования. При создании обобщенного сценария для тестирования интерфейса пользователя необходимо выполнить следующие шаги:

- 1) запись сценария с использованием штатных инструментов тестирования на одном из программных комплексов;
- 2) определение уникальных для всех программных комплексов последовательностей действий;
- 3) в сгенерированных Unit-тестах для уникальных последовательностей действий вынесение с использованием рефакторинга фрагментов Java-кода в класс на базе `TestClientScenario`;
- 4) в случае отличий в поведении серверной части описание соответствующей логики на базе `TestServerScenario`;
- 5) выделение данных, необходимых для исполнения сценария, и помещение их в классы `TestData`;
- 6) формирование `TestClientScenario` и `TestData` для остальных программных комплексов с учетом их особенностей.

Выполнение шага 4 требует от тестировщика не только хорошего знания инструментов автоматизированного тестирования, программирования на Java и умения писать Unit-тесты, но и знания логики построения серверной части, что в общем случае известно только разработчику. Целесооб-

разным представляется данный шаг выполнять авторам серверной части приложений.

Указанную последовательность шагов необходимо повторить столько раз, сколько обобщенных сценариев необходимо создать для полноценного тестирования всех программных комплексов обработки измерительной информации.

В процессе выполнения тестов по обобщенным сценариям на основании указанной структуры классов будут автоматически формироваться уникальные сценарии тестирования для конкретного программного комплекса, учитывающие как особенности построения клиентской и серверной части, так и входных и выходных данных.

Преимущества обобщенных сценариев:

- при наличии обобщенного сценария, созданного для одного из программных комплексов, формирование сценариев для остальных программных комплексов является нетрудоемкой задачей;
- в случае изменений и исправления ошибок в общей части программных комплексов изменяется общая часть сценариев тестирования, что сокращает время внесения изменений;
- при появлении новых аналогичных программных комплексов сложность создания сценариев существенно меньше, чем при написании их с самого начала;
- при появлении новой функциональности требуются незначительные изменения обобщенного сценария.

Недостатки обобщенных сценариев:

- существенное время разработки первого обобщенного сценария, которое может окупиться только при условии его повторного использования;
- высокая сложность выделения общих частей, которая требует соответствующей квалификации тестировщика;
- объем обобщенного сценария в общем случае больше обычного сценария.

В результате преимущества обобщенных сценариев заключаются в возможности их повторного использования для идентичных программных комплексов, что значительно снижает трудоемкость разработки новых сценариев. Данный подход с учетом указанных особенностей применим для программных комплексов обработки измерительной информации, но может быть недостаточно эффективен при выполнении автоматизированного тестирования в других областях.

СПИСОК ЛИТЕРАТУРЫ

Рефакторинг: улучшение существующего кода / М. Фаулер, К. Бек, Д. Брант и др. М.: Символ-Плюс, 2009.

O. S. Chernaya, Yu. Yu. Fedorova, S. A. Belyaev

APPLYING OF METHODS AND TOOLS TO AUTOMATED TESTING OF TELEMETRIC DATA PROCESSING SOFTWARE

The method of common automated testing scenarios design, classes structure for such scenarios, advantages and lacks are described in the article.

Automated testing, common scenarios, WindowTester Pro

УДК 681.3, 681.5

М. С. Куприянов, Ю. А. Шичкина

Схема укрупнения операций распараллеливаемого алгоритма

Рассматривается построение информационного графа по последовательному алгоритму и его преобразование к параллельной форме с достаточной шириной ярусов и с последующей реализацией на параллельной вычислительной среде. Показано также, что формализация данного процесса позволяет найти оптимальное решение задачи распараллеливания алгоритма с учетом таких параметров, как количество процессоров, время вычислений и плотность вычислений на единицу времени.

Параллельный алгоритм, информационный граф, ширина графа, список следования

Практически одновременно с появлением компьютера человечеству стало ясно, что созданные компьютеры не в состоянии решить за приемлемое время многие задачи. Несмотря на то, что производительность компьютеров росла экспоненциально начиная с 1945 г. и продолжает расти в настоящее время, потребность в увеличении скорости обработки данных превышает показатели этого роста.

Параллельное программирование, как и обычное программирование, опирается на алгоритмы и структуры данных. Неэффективно работающая программа – это прямые потери производительности компьютера, средств на его приобретение, усилия на освоение и т. п. Таких потерь хотелось бы избежать или, по крайней мере, их минимизировать. Для этого необходимы апробированные методика анализа алгоритмов и методы их оптимизации.

Чаще всего оптимизация параллельных алгоритмов рассматривается в условиях неограниченного параллелизма, которые предполагают использование идеализированной модели парал-

лельной вычислительной системы и которых на практике не существует.

Эффективность параллельного алгоритма зависит от многих параметров. Одним из них является равномерная загрузка процессоров вычислительной системы. При этом важно, с одной стороны, сохранив высоту параллельного алгоритма, минимизировать его ширину. Эту проблему можно решить с помощью метода, представленного в [1]. Другим важным параметром является время выполнения процессов. Если предполагать, что время выполнения всех процессов, отраженных в информационном графе в виде отдельных вершин, одинаково, тогда ориентированный ациклический информационный граф является наиболее подходящим инструментом для построения оптимального по высоте и ширине параллельного алгоритма. Но на практике такие условия – крайняя редкость. Следовательно, из-за неравномерности выполнения отдельных процессов часть процессоров будет простаивать в ожидании результатов от своих предшественников. Поэтому сама собой напрашивается замена ориентирован-

ного ациклического информационного графа его взвешенным аналогом, в котором в качестве весов будет выступать время выполнения процессов.

Далее предлагается метод, позволяющий оптимизировать параллельный алгоритм по времени при сохранении или уменьшении числа вычислительных единиц.

Метод укрупнения операций алгоритма по взвешенному информационному графу. Временной список следования – совокупность элементов, полученных на основе матрицы следования по правилу: если на пересечении i -й строки и j -го столбца матрицы смежности стоит единица, то номера этих строки и столбца образуют в списке элемент (v_i, v_j, t_j) , где t_j – время выполнения операции v_j .

Список следования позволяет сэкономить память при хранении информации о взвешенном графе и увеличить скорость обработки данных при анализе алгоритма на существование просторов процессоров при его реализации.

Порядок выполнения операций следующий:

1. Рассчитать значение теоретической минимальной ширины d информационного графа [1].

2. Добавить в граф вершину. Все выходные вершины соединить с новой вершиной. Таким образом, в графе будет всего одна выходная $(n+1)$ -я вершина.

3. Построить соответствующий информационному графу временной список следования $V_0 = (V_1, V_2, t)$.

4. Построить множество выходных вершин V_k .

5. Найти множество $V = V_2 - V_1$. Вершины, вошедшие в это множество, составят группу вершин M_i , принадлежащих одному i -му ярусу.

• Если число вершин множества $V = V_2 - V_1$ превышает d , то возможны следующие варианты:

а) во множестве M_i существует ровно d вершин с одинаковым временем выполнения. В данном случае эти d вершин составят множество M_i ;

б) во множестве M_i существует больше чем d вершин с одинаковым временем выполнения. В этом случае любые d вершин с одинаковым временем выполнения составят множество M_i ;

в) во множестве M_i существует менее d вершин с одинаковым временем выполнения. В этом

случае множество M_i составят d вершин с минимальным временем выполнения.

Остальные вершины необходимо включить в следующую группу M_{i+1} . При повторном проходе этого шага в группу M_{i+1} будут добавлены вершины.

• Если число вершин множества $V = V_2 - V_1$ меньше числа d , то во множество M_i следует перенести вершины из множества V_k , удовлетворяющие правилу: $V_{kj} \cap V_1 = \emptyset$, т. е. вершины из V_k , отсутствующие во множестве V_1 . Эти вершины могут быть взяты в произвольном порядке из множества: $V_k - V_1$.

6. Удалить из списка смежности все пары, конечная вершина которых (V_2) совпадает с одной из вершин множества V , и построить тем самым список V_i .

7. Если $i = 1$, то вернуться на шаг 5.

8. Во множестве M_{i-1} найти вершину с минимальным временем выполнения m_{i-1k} . Если все вершины имеют одинаковое время выполнения, то перейти к шагу 16.

9. Составить множества:

– $S_{m_{i-1k}M_i}$ – совокупность вершин из множества M_i , связанных с вершиной m_{i-1k} одним ребром и являющихся для данного ребра конечными вершинами;

– $S_{(M_{i-1}-m_{i-1k})M_i}$ – совокупность вершин из множества M_i , связанных с вершинами множества M_{i-1} (за исключением вершины m_{i-1k}) одним ребром и являющихся для данного ребра конечными вершинами.

Внимание! Если во множестве M_{i-1} есть укрупненные операции, то рассматривается всегда только последняя добавленная при укрупнении вершина.

10. Найти разность: $S = S_{m_{i-1k}M_i} - S_{(M_{i-1}-m_{i-1k})M_i}$.

• Если $S = \emptyset, S_{m_{i-1k}M_i} \neq \emptyset, S_{(M_{i-1}-m_{i-1k})M_i} \neq \emptyset$, то перейти к шагу 5.

• Если $S_{m_{i-1k}M_i} = \emptyset, S_{(M_{i-1}-m_{i-1k})M_i} = \emptyset$, то $S = M_i$.

• Если $S_{m_{i-1k}M_i} = \emptyset, S_{(M_{i-1}-m_{i-1k})M_i} \neq \emptyset$, то $S = M_i - S_{(M_{i-1}-m_{i-1k})M_i}$.

№	V_1	V_2	t
1	6	8	3
2	9	8	3
3	4	3	3
4	12	8	3
5	12	9	2
6	10	8	3
7	10	4	2
8	10	3	3
9	10	9	2
10	10	6	1
11	7	8	3
12	7	4	2
13	7	3	3
14	7	9	2
15	7	6	1
16	7	10	1
17	11	8	3
18	11	4	2
19	11	3	3
20	11	9	2
21	11	6	1
22	11	10	1

№	V_1	V_2	t
23	14	8	3
24	14	4	2
25	14	3	3
26	14	9	2
27	14	6	1
28	14	10	1
29	14	7	2
30	5	4	2
31	5	3	3
32	5	1	2
33	2	1	2
34	15	1	2
35	15	2	1
36	13	12	3
37	16	8	3
38	16	4	2
39	16	3	3
40	16	9	2
41	16	6	1
42	16	10	1
43	16	13	1
44	16	7	2

№	V_1	V_2	t
45	16	1	2
46	16	14	2
47	16	2	1
48	16	12	3
49	16	15	3
50	17	1	2
51	17	2	1
52	17	3	3
53	17	4	2
54	17	5	2
55	17	6	1
56	17	7	2
57	17	8	3
58	17	9	2
59	17	10	1
60	17	11	3
61	17	12	3
62	17	13	2
63	17	14	2
64	17	15	3
65	17	16	1

Рис. 1

11. Найти во множестве вершину с минимальным временем выполнения $s_{\min}$. Удалить эту вершину из множества M_i . Добавить эту вершину во множество M_{i-1} , укрупнив операцию m_{i-1k} слиянием ее с операцией $s_{\min}$:

$$m_{i-1k} = m_{i-1k} \cup s_{\min}, t_{i-1k} = t_{i-1k} + t_{s_{\min}}.$$

12. Если $M_i = \emptyset$, то удалить множество $M_i = \emptyset$ и сдвинуть счетчик множеств $i = i - 1$. Перейти к шагу 4. Если $M_i \neq \emptyset$, то перейти к шагу 9.

13. Если список не пустой, то вернуться на шаг 4.

14. Если в списке не осталось ни одной пары, работа окончена.

Общее время работы алгоритма составит

$$T = \sum_{i=1}^k t_{i,\max}, \text{ где } k - \text{число групп; } t_{i,\max} - \text{максимальное время среди операций в } i\text{-й группе.}$$

Пример. Первоначальный список следования, соответствующий некоторому информационному графу, представлен на рис. 1.

Найдем разность множеств $V = V_2 - V_1 = \{1, 3, 8\}$.

Эта группа вершин составит первый ярус информационного графа: $M_1 = \{1, 3, 8\}$ (рис. 2).

M_1	
i	t
1	2
3	3
8	3

Рис. 2

Теоретическая минимальная ширина графа равна 3, поэтому в данную группу пока не будем добавлять вершины из конечной группы V_k . Удалим из первоначального списка все пары, вторым элементом которых является одна из вершин группы M_1 , и получим новый список. Вновь найдем разность множеств конечных и начальных вершин ребер графа $V = V_2 - V_1 = \{9, 4, 6, 2\}$ (рис. 3).

M_2	
j	t
9	2
4	2
6	1
2	1

Рис. 3

Во множестве M_1 найдем вершину с минимальным временем выполнения $m_{11} = 1, t_1 = 2$. Составим множества S_{1M_2} (совокупность вершин из множества M_2 , связанных с вершиной 1 одним ребром и являющихся для данного ребра конечными вершинами) и $S_{\{3,8\}M_2}$ (совокупность вершин из множества M_2 , связанных с вершинами множества M_1 , за исключением вершины 1, одним ребром и являющихся для данного ребра конечными вершинами). Найдем разность: $S = S_{1M_2} - S_{\{3,8\}M_2}$ (рис. 4).

S_{1M_2}	$S_{\{3,8\}M_2}$	$S = S_{1M_2} - S_{\{3,8\}M_2}$
4	4	2
6	9	
2	6	

Рис. 4

Во множестве S одна вершина 2. Удалим эту вершину из множества M_2 и добавим ее во множество M_1 , укрупнив тем самым операцию 1 слиянием ее с операцией 2: $1' = \{1, 2\}, t_1 = t_1 + t_2 = 2 + 1 = 3$ (рис. 5).

Скорректированные группы					
M_1			M_2		
i	t		j	t	
1,2	3		9	2	
3	3		4	2	
8	3		6	1	

Рис. 5

В результате, первая группа выровнена по времени. На этом ее фиксируем. Дальнейшим изменениям данная группа не подлежит. Продолжая далее, получим в итоге 3 группы (рис. 6).

M_1		M_2		M_3	
i	t	j	t	b	t
1,2	3	9,12,13,14	8	16	1
3	3	4,10,5,11	8		
8	3	6,15,7	6		

Рис. 6

Общее время работы алгоритма

$$T = \sum_{i=1}^k t_{i,\max} = 3 + 8 + 1 = 12.$$

Проиллюстрируем результаты с помощью временных диаграмм.

До применения метода оптимизации по времени с помощью метода оптимизации по ширине

[1] были получены 6 групп операций для трех процессоров (рис. 7).

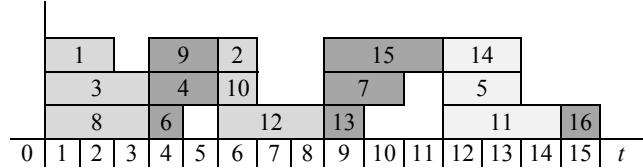


Рис. 7

Время работы алгоритма равно 15 ед.

После применения метода оптимизации по времени с сохранением минимальной теоретической ширины $d = 3$ было получено 3 группы (рис. 8).

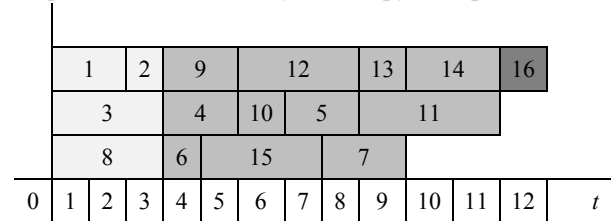


Рис. 8

Время работы алгоритма равно 12 ед.

Анализ трудоемкости показал, что применение временных списков следования более эффективно по сравнению с аналогичным методом, состоящим из двух частей [2] и основанным на матрице смежности. Метод оптимизации информационного графа по времени с применением временных списков следования позволяет значительно сократить время работы алгоритма с сохранением заданной ширины графа. При этом ширина графа может быть оговорена произвольным образом: либо на основании найденной теоретической минимальной ширины графа, либо в соответствии с особенностями вычислительной системы.

Следует заметить, что решение задачи минимизации алгоритма по времени вычисления может быть не единственным. Применение изложенного метода оптимизации информационного графа по времени с использованием временных списков следования позволяет свести время выполнения алгоритма к минимальному при заданной ширине и найти только одно из решений задачи минимизации. Так, в приведенном примере существует еще одно решение (рис. 9).

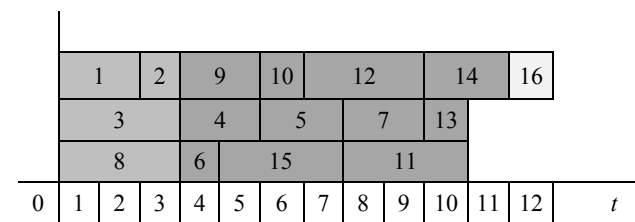


Рис. 9

Время работы алгоритма равно 12 ед.
Работа выполнена в СПбГЭТУ при финансо-
вой поддержке Министерства образования и науки

Российской Федерации (договор № 02.G25.31.0058
от 12.02.2013).

СПИСОК ЛИТЕРАТУРЫ

1. Шичкина Ю. А. Сокращение высоты информа-
ционного графа параллельного алгоритма // Науч.-
технические ведомости СПбГПУ. Информатика. Теле-
коммуникации. Управление. 2009. № 3 (80). С. 148–152.

2. Шичкина Ю. А. Применение списков следования
для оптимизации информационного графа по высоте
// Системы. Методы. Технологии. 2011. № 1 (9). С. 68–77.

M. S. Kupriyanov, Yu. A. Shichkina

SCHEME OF OPERATIONS ENLARGING FOR PARALLELIZED ALGORITHMS

One of the solutions of the problem designing the effective numerical methods for parallel architecture computers is building the information graph by serial algorithm and transforming by list consecutions method to parallel form with sufficient edge width and implementation with parallel system. This article points out that it is possible not only to formalize this process but also to find out the optimal solution for parallelizing the algorithm with a glance to such parameters as a number of processor units, computation time and computation density per unit time.

Parallel algorithm, information graph, width of graph, list consecutions



УДК 512.2

А. В. Васильев, Н. А. Жукова

Методы оценивания информативности результатов опытной эксплуатации сложных многопараметрических объектов в условиях неопределенности

Предлагается метод оценивания информативности результатов испытательных работ, реализуемых на этапе опытной эксплуатации сложных технических объектов, в условиях различного уровня неопределенности информации о параметрах объектов и степени их влияния на оценку информативности работы в целом.

Интеллектуальный анализ данных, символьное представление, телеметрические параметры

Эксплуатация сложных технических объектов сопровождается непрерывным контролем, осуществляемым на основе обработки и анализа передаваемой с объектов телеметрической информации (ТМИ) [1]. С помощью анализа телеметрической информации решается широкий круг задач, включающий задачи исследования поведения телеметрируемого объекта, определения его технического состояния, а также локализации возникающих неисправностей. К группе задач исследования поведения объекта относятся определение характеристик и идентификация объекта. В набор определяемых по ТМИ характеристик входят в том числе характеристики, определяющие основные функциональные и технические возможности объекта. Ценность телеметрической информации определяется объектом, функциональные возможности которого она раскрывает.

В области эксплуатации ракетно-космической техники (РКТ) имеет большое значение определение информативности, полученной в ходе проведенных пусков объектов различных типов. В связи с высокой сложностью объектов РКТ они характеризуются большим количеством разнородных плохо формализуемых параметров, имеющих различную информационную ценность. Оценка информативности ТМИ, полученной в результате проведения пуска, на основе совокупности заре-

гистрированных параметров без использования специализированных методов и средств требует существенных временных затрат и может привести к формированию ошибочных результатов. Дополнительная сложность, возникающая при оценке информативности пусков РКТ, связана с отсутствием методов и алгоритмов определения информативности отдельных параметров. Оценки одной ТМИ, получаемые разными экспертами, могут существенно различаться, при этом ни один из результатов не может считаться обоснованным, поскольку построен на экспертных знаниях – нечетких и, как следствие, плохо формализуемых.

В статье предлагается метод оценивания информативности результатов пусков РКТ, реализуемых на этапе их опытной эксплуатации, в условиях различного уровня неопределенности информации о параметрах объектов и степени их влияния на оценку информативности пуска в целом.

Анализ состава и характеристик параметров, получаемых в результате проведения пуска. Оценка информативности пусков РКТ может проводиться на различных этапах испытаний. Так, может оцениваться информативность планируемого испытания, при этом полученные результаты сравниваются с ранее накопленными характеристиками информативности, что позволяет

выделить составляющие показатели испытываемого образца РКТ, наиболее важные для общей оценки результатов испытаний, и правильно поставить задачу персоналу на предстоящий пуск.

Оценка результатов проведенных испытаний с точки зрения интегральной характеристики информативности позволяет определить степень реализации задач пуска и дополнить общую статистику информационного обеспечения испытаний. Такие интегральные оценки информативности также позволяют корректно спланировать задачи последующих испытаний для повышения значимости планируемого испытания. Так, мало-значительные для общей оценки характеристик объекта испытаний показатели информативности могут быть исключены из целей предстоящего испытания, а наиболее ценным показателям может быть уделено дополнительное внимание.

Показателями информативности могут, например, служить:

- 1) тип ракеты;
- 2) цель полета (испытание вооружений или вывод полезной нагрузки);
- 3) количество ранее проведенных испытаний подобных ракет;
- 4) является ли данная ракета самостоятельной разработкой или модернизацией существующего типа;
- 5) наличие нового типа бортовой аппаратуры ТМИ;
- 6) отличие циклограммы полета ракеты от типовой;
- 7) отличие реализованной траектории полета от типовой;
- 8) отличие района запуска от типового;
- 9) отличие района прицеливания от типового;
- 10) отличие времен отделения ступеней от типового;
- 11) факт отработки модернизированного оснащения;
- 12) факт отработки модернизированных ступеней;
- 13) наличие аварийной ситуации на борту;
- 14) объем и качество полученной измерительной информации.

Состав показателей может меняться и дополняться в зависимости от хода испытаний и их конкретных текущих задач.

Получив характеристики интегральной информативности планируемого испытания и проведя сравнение ее с оценками ранее проведенных

пусков, становится возможно корректировать программы испытания с целью исключения задач, получение результатов по которым мало изменит общие оценки характеристик испытываемого образца РКТ. Это позволит снизить материальные и людские затраты на проведение дорогостоящих летных испытаний РКТ и повысит общую эффективность этого этапа создания новых образцов.

Обобщенный метод оценивания информативности результатов (ИР) опытной эксплуатации сложных многопараметрических объектов в условиях неопределенности. В соответствии с типовыми ситуациями, возникающими при решении практических задач, выделяют три уровня неопределенности:

– первый уровень соответствует наличию у эксперта предметной области информации о составе параметров, характеризующих объект испытаний, информативности параметров, значениях всех параметров объектов, полученных в результате проведения испытательных работ;

– второй уровень говорит о наличии информации о составе параметров, характеризующих объект испытаний, их информативности и значениях, полученных при выполнении работ, при этом степень влияния отдельных параметров на формируемый результат четко не определена;

– третий уровень свидетельствует только о наличии информации о составе параметров и их значениях. Данные об информативности параметров и степени их влияния на информативность пуска отсутствуют.

Для каждого уровня неопределенности предлагается применять специализированный алгоритм, учитывающий объем информации, имеющийся у эксперта соответствующего уровня.

Метод оценки информативности результатов опытной эксплуатации в условиях наличия полной информации о результатах проведенных работ. При наличии полной информации о результатах испытательных работ и степени информативности отдельных параметров для оценки работы в целом достаточно учесть имеющиеся данные.

Метод 1. Метод оценки информативности результатов опытной эксплуатации в условиях наличия полной информации.

Вход: $K = k_1, \dots, k_n$ – множество параметров объектов испытаний; $V = v_1, \dots, v_n$ – весовые коэффициенты параметров объектов; $O = o_1, \dots, o_m$ – множество объектов испытаний; $Q = \{q_{ij}\}$, $i = 0, \dots, m$, $j = 0, \dots, n$ – значения параметров объектов.

Выход: $I = \{i_i\}$, $i = 0, \dots, m$ – показатели информативности работ.

1: для $i = 1$ до n

2: $V' \leftarrow$ выполнить нормализацию весовых коэффициентов

$Q' \leftarrow$ выполнить нормализацию значений параметров

3: для $i = 1$ до m

4: вычислить информативность каждой из работ в соответствии с выражением

$$I_j = \sum_{i=1}^n Q_{ij} \times V_i$$

5: выполнить ранжирование работ в соответствии с информативностью I

Входные данные, необходимые для работы метода, представлены в виде таблицы (табл. 1).

Таблица 1

Перечень испыта- тельных работ	Перечень параметров объекта испытаний								Показа- тель информа- тивности испыта- тельной работы
	K_1	K_2	K_3	...	K_i	...	K_{n-1}	K_n	
	Показатели информативности (ве- совые коэффициенты) параметров								
	V_1	V_2	V_3	...	V_i	...	...	V_n	
ИР 1 / Объект 1	Q_{11}	Q_{21}	Q_{31}	...	Q_{i1}	...	...	Q_{n1}	I_1
ИР 2 / Объект 2	Q_{12}	Q_{22}	Q_{32}	...	Q_{i2}	...	...	Q_{n2}	I_2
ИР 3 / Объект 3	Q_{13}	Q_{23}	Q_{33}	...	Q_{i3}	...	...	Q_{n3}	I_3
...									
...									
ИР K / Объект K	Q_{1j}	Q_{2j}	Q_{3j}	...	Q_{ij}	...	...	Q_{nj}	I_j
...									
...									
ИР M / Объект M	Q_{1m}	Q_{2m}	Q_{3m}	...	Q_{im}	...	...	Q_{nv}	I_m

В таблице используются следующие основные обозначения: K_i – i -й параметр объекта испытаний; V_i – показатель информативности (весовые коэффициенты) i -го параметра объекта испытаний; Q_{i1} – значение i -го параметра j -го объекта испытаний, полученного при проведении j -й работы. При определении набора параметров объектов предусматривается возможность указать выделенные параметры K^* . Когда значение выделенного параметра равно единице, показателю информативности работы присваивается максимальное значение.

Метод предполагает выполнение трех основных этапов – нормирование исходных данных, задаваемых пользователем; вычисление информативности работ; ранжирование выполненных работ в соответствии с их информативностью.

Метод оценки ИР опытной эксплуатации в условиях отсутствия части информации о результатах проведенных работ. При отсутствии информации о степени влияния отдельных параметров на формируемый результат необходимо решить задачу определения функциональной зависимости значений каждого параметра и информативности результатов работ. Функциональные зависимости строятся для каждого параметра и представляются в виде отдельных показателей, описывающих их поведение.

Отдельный показатель g_i , описывающий параметр k_i , представляет собой невозрастающую или неубывающую функцию $g_i = g_i(k_i)$, $i = 1, \dots, m$. Отдельный показатель g_i характеризует степень проявления оцениваемого i -го отдельного параметра у объектов. При этом значение $g_i^{(j)} = g_i(k_i^{(j)}) = 0$ указывает на полное отсутствие проявления i -го параметра j -го объекта и, как следствие, параметр не оказывает влияния на информативность j -й работы, а значение $g_i^{(j)} = g_i(k_i^{(j)}) = 1$ – на максимально возможное проявление данного параметра у j -го объекта. Для задания функции $g_i = g_i(k_i)$ эксперт определяет значения MIN_i и MAX_i параметра k_i , указывающие верхние и нижние границы областей, в которых отдельный показатель q_i принимает значения 0 или 1 соответственно, а также определяет вид функции $g_i(k_i)$. Для описания отдельных показателей рекомендуется использовать степенную функцию, в этом случае эксперт указывает степень функции P_i , задающую направление и степень вогнутости функции $g_i(k_i)$ на отрезке $[\text{MIN}_i, \text{MAX}_i]$. Для работы с бинарными и качественными параметрами объектов формируется единая порядковая шкала, к которой приводятся значения всех параметров.

Далее приводится описание метода оценки информативности результатов опытной эксплуатации в условиях отсутствия части информации, основанного на формировании отдельных показателей для параметров объектов.

Метод 2. Метод оценки информативности результатов опытной эксплуатации в условиях отсутствия части информации.

Вход: $K = k_1, \dots, k_n$ – множество параметров объектов испытаний; $V = v_1, \dots, v_n$ – весовые коэффициенты параметров объектов; $O = o_1, \dots, o_m$ – множество объектов испытаний; $Q = \{q_{ij}\}$, $i = 0, \dots, m$, $j = 0, \dots, n$ – значения параметров объектов.

Выход: $I = \{i_i\}$, $i = 0, \dots, m$ – показатели информативности работ.

1: для $i = 1$ до

2: определить $g_i(k_i)$

3: для $i = 1$ до n

4: для $i = 1$ до m

5: если значение q_{ij} не определено

6: $q_{ij} = g_i(k_i^{(j)})$

7: $I \leftarrow$ результаты выполнения метода 1

В состав метода входят два этапа: формирование отдельных показателей и выполнение метода 1.

Метод оценки ИР опытной эксплуатации в условиях отсутствия части информации о результатах проведенных работ, а также информации об информативности отдельных параметров. Значения весовых коэффициентов в условиях отсутствия априорной информации об информативности параметров предлагается определять с использованием метода оценки весовых коэффициентов, основанных на рандомизированном выборе коэффициентов. Метод предложен в рамках метода вычисления сводных показателей [2].

Предполагается, что компоненты вектора весовых коэффициентов $\mathbf{v} = (v_1, \dots, v_n)$ отсчитываются дискретно с шагом $h = 1/w$, где w – число градаций значимости отдельных показателей, т. е. весовые коэффициенты принимают значения из множества $\{0, 1/h, 2/h, \dots, (h-2)/h, (h-1)/h, 1\}$. Неопределенность выбора вектора $\mathbf{v} = (v_1, \dots, v_n)$ из множества вариантов $W(m, n)$ моделируется рандомизацией этого выбора.

Для сокращения множества $W(m, n)$ всех возможных векторов весовых коэффициентов в [2] предлагается учитывать нечисловую и неточную информацию о весовых коэффициентах, которая задается пользователем с помощью системы равенств и неравенств вида $v_i = v_j$, $v_r > v_s$, $i, j, r, s \in \{1, \dots, n\}$, а также неравенств вида $a_i \leq v_i \leq b_i$, $i = 1, \dots, n$, где $0 \leq a_i \leq b_i \leq 1$. При

наличии дополнительной информации весовые коэффициенты представляют собой случайные величины $\tilde{v}_1(I), \dots, \tilde{v}_m(I)$, имеющие совместное равномерное распределение на множестве допустимых с точки зрения информации I весовых векторов $W(m, n; I)$. В качестве числовых оценок $\bar{v}_i(I)$ весовых коэффициентов, удовлетворяющих равенствам и неравенствам системы I , рекомендуется использовать математические ожидания $E\tilde{v}_i(I)$ рандомизированных весовых коэффициентов $\tilde{v}_i(I)$, $i = 1, \dots, m$, образующих случайный весовой вектор $\tilde{\mathbf{v}}(I) = (\tilde{v}_1(I), \dots, \tilde{v}_m(I))$. Точность оценок определяется при помощи стандартных отклонений $s_1(I), \dots, s_m(I)$, соответствующих случайным весовым коэффициентам.

При вычислении информативности выполненных работ на этапе оценки значений весовых коэффициентов возможно возникновение ситуации, при которой эксперт может задать значения для части весовых коэффициентов; при этом значения неизвестных коэффициентов необходимо доопределить, для чего применяется рассмотренный метод. При этом заданные экспертом коэффициенты рассматриваются как константные.

Метод оценки ИР опытной эксплуатации в условиях отсутствия части информации о результатах проведенных работ и отсутствии информации об информативности отдельных параметров предполагает вычисление весовых коэффициентов параметров объектов, а затем выполнение метода 2. Далее приводится описание рассмотренного метода.

Метод 3. Метод оценки информативности результатов опытной эксплуатации в условиях отсутствия части информации о результатах проведенных работ, а также информации об информативности отдельных параметров.

Вход: $K = k_1, \dots, k_n$ – множество параметров объектов испытаний; $O = o_1, \dots, o_m$ – множество объектов испытаний; $Q = \{q_{ij}\}$, $i = 0, \dots, m$, $j = 0, \dots, n$ – значения параметров объектов.

Выход: $I = \{i_i\}$, $i = 0, \dots, m$ – показатели информативности работ.

1: сформировать вектор весовых коэффициентов V

2: $I \leftarrow$ результаты выполнения метода 2

Предложенный метод позволяет обеспечить поддержку принятия решений экспертами при оценивании информативности результатов опытной эксплуатации сложных многопараметрических объ-

ектов в условиях различного уровня неопределенности. Применение метода позволяет формализовать и интегрировать неполные и нечеткие знания различных экспертов и использовать их для обоснования формируемых решений.

Рассмотренный метод был апробирован и показал свою эффективность при выполнении опытно-конструкторской работы, посвященной вопросам анализа информативности результатов пусков групп объектов различных типов.

СПИСОК ЛИТЕРАТУРЫ

1. Назаров А. В. Современная телеметрия в теории и на практике: Учеб. курс. СПб.: Наука и техника, 2007. 672 с.

2. Хованов Н. В. Анализ и синтез показателей при информационном дефиците. СПб.: Изд-во Санкт-Петерб. ун-та, 1996. 196 с.

A. V. Vasiljev, N. A. Zhukova

METHODS FOR INFORMATIONAL CONTENT ESTIMATION OF RESULTS OF DIFFICULT MULTIPLE PARAMETER OBJECTS TRIAL OPERATION IN THE CONDITIONS OF UNCERTAINTY

The method for estimation of informational content of the test works results of difficult technical objects performed at a stage of trial operation for various levels of information uncertainty about objects parameters and extent of their influence on an assessment of informational content of the whole work is offered.

Information system architecture Intellectual data analysis, symbolic representation, telemetric parameters

УДК 004.41

А. А. Лисс, А. С. Скрипникова

Автоматизированная система архивирования и аудита электронного обмена в рабочих процессах таможенного дела

Приводится информация об автоматизированной системе, разработанной в рамках проекта модернизации информационно-технического обеспечения таможенных органов Российской Федерации. Обсуждаются функциональные возможности системы по архивированию и аудиту сообщений электронного обмена в процессах таможенной деятельности. Подчеркивается возможность использования системы для выявления закономерностей электронного обмена и выбора тактики реагирования на нештатные ситуации.

Автоматизированная система таможенных органов, электронное представление сведений, электронное сообщение, информационный обмен, нештатная ситуация, восстановление данных

С начала января 2011 по июнь 2013 г. Федеральной таможенной службой Российской Федерации (ФТС России) проводились работы по комплексной модернизации информационно-технического обеспечения таможенных органов Российской Федерации, предусматривающей совершенствование нормативно-правовой базы, рабочих процессов и информационных технологий. Проект был предпринят с целью развития внешней тор-

говли Российской Федерации, совершенствования таможенного администрирования, создания эффективной системы таможенного контроля, упрощения таможенных операций и, как следствие этого, увеличения внешнеторгового оборота и собираемости таможенных платежей. В результате модернизации создан единый комплекс функциональных систем и подсистем, взаимодействующих друг с другом и с внешними автомати-

зированными информационно-программными средствами (ИПС) в основном посредством обмена электронными сообщениями.

В рамках указанного проекта открытым акционерным обществом «Научно-инженерный центр Санкт-Петербургского электротехнического университета» (ОАО «НИЦ СПб ЭТУ») разработана автоматизированная система архивирования и аудита электронного обмена в рабочих процессах таможенного дела, получившая рабочее название ПАВИС. Необходимость разработки ПАВИС была продиктована приоритетными требованиями ФТС России к обеспечению высокого уровня надежности электронного обмена, возможности проведения круглосуточного мониторинга и анализа нештатных ситуаций, возникающих в процессе электронного обмена, а также возможности быстрого восстановления сообщений, потерянных в результате возможных аварий и сбоев. Поэтому основным назначением ПАВИС является гарантированное хранение всех сообщений, формируемых при взаимодействии компонентов модернизированной системы таможенных органов и обеспечение регламентированного доступа к ним. В свою очередь, эти возможности позволяют с помощью ПАВИС проводить анализ закономерностей электронного обмена и принимать решения относительно предпочтительной тактики реагирования на нештатную ситуацию. Назначением системы определяется круг ее пользователей. Для ПАВИС это – должностные лица, в обязанности которых входит мониторинг работоспособности системы, анализ ошибок и отклонений от установленных регламентов, аудит информационной безопасности и конструктивное устранение последствий нештатных ситуаций. Рассмотрим более подробно функциональные возможности системы.

Большинство ИПС взаимодействуют между собой в транспортной технологической подсистеме таможенных органов (ТПП) с использованием сетевого программного обеспечения IBM WebSphere MQ. Для обмена создается технологический конверт, соответствующий определенным требованиям и состоящий из адресной части и вложенных документов в XML-формате. Такой способ обмена гарантирует доставку сообщений системе-получателю даже в случае сбоев в функционировании ТПП. Доставка сообщений не связана с протоколами пакетной передачи данных, что позволяет обеспечивать асинхронность до-

ставки, и не требует тесной связи ИПС. В модернизированной автоматизированной системе таможенных органов в обмене участвует также автоматизированная система интеграции бизнес-процессов (АС ИБП), выполняющая, в частности, функцию маршрутизации. Причем все сообщения от систем-отправителей, поступающие на вход АС ИБП, средствами IBM WebSphere MQ направляются также на вход ПАВИС.

При получении входного сообщения ПАВИС сохраняет его в базе данных и выполняет автоматический контроль сохраняемого сообщения. Контроль сообщений содержит несколько уровней. Сначала выполняется структурный контроль, позволяющий проверить, корректное ли XML-сообщение. Далее проверяется целостность сообщения, которая гарантируется достоверностью электронной подписи сообщения. Завершают контроль этапы проверки структурного регламента и регламента движения сообщений. В процессе контроля структурного регламента проводится проверка, соответствует ли сообщение утвержденным для системы-отправителя спецификации обмена и структуре (XML-схеме), а при контроле регламента движения сообщений определяется, укладывается ли время получения и обработки сообщения системой-получателем в заданные временные рамки. При проверке регламента движения анализируется информация, содержащаяся в технологических квитанциях о получении и обработке сообщения системой-получателем. Параметры проверок структурного регламента и регламента движения настраиваются в ПАВИС должностным лицом, уполномоченным для работы в роли оператора по настройкам и регламентам. ПАВИС предоставляет пользователям удобный графический интерфейс для ввода, просмотра и редактирования информации о регламентах, а для упрощения работы предлагается «пошаговый мастер» создания регламентов.

Если новый регламент незначительно отличается от одного из существующих, то регламент, созданный ранее, можно использовать как шаблон. При этом система фиксирует все внесенные изменения, формируя журнал изменений, в котором отображается имя зарегистрированного в системе пользователя, внесшего изменения, дата и время внесения изменений, их содержание. Записи в журнале выводятся: красным цветом – для удаленных элементов, желтым – для элементов, в которые внесены изменения, и зеленым – для добав-

ленных элементов с целью отображения характера изменения. При необходимости текст регламента, перечень регламентов, а также журнал изменений можно вывести на печать.

Если сообщение не прошло хотя бы один из уровней контроля, должностному лицу, выполняющему мониторинг работоспособности автоматизированной системы таможенных органов, поступает информация о факте обнаружения ошибки в процессе обмена. Первичный анализ для локализации проблемы выполняется с помощью программной задачи «Представление и анализ информации», входящей в состав ПАВИС. Инструментальные средства программной задачи позволяют определить, на каком уровне контроля была выявлена ошибка. Если она была выявлена при проверке целостности сообщения, это может сигнализировать о том, что в процессе доставки в сообщение были внесены изменения. Поскольку ПАВИС специально спроектирована для хранения сообщений в виде, в котором их формирует система-отправитель, существует возможность сравнить сообщение, сохраненное в ПАВИС, с сообщением, поступившим на вход системы-получателя.

Кроме первичного анализа причины ошибки программная задача «Представление и анализ информации» предоставляет полезную возможность анализировать потоки сообщений и на основе результатов анализа вырабатывать рекомендации для оптимизации взаимодействия. Для этого предусмотрены средства выборки массивов сообщений по заданным параметрам, просмотра информации о сообщении, включая результаты его обработки при получении, вывода графиков загрузки на основе временных параметров обмена.

Если ошибка была выявлена при структурном контроле или проверке регламентов, анализ выполняется в программной задаче «Анализ ошибок и несоответствий». В программной задаче предусмотрена возможность поиска сообщений об ошибке в базе данных, содержащей их описания, с использованием достаточного набора критериев. Например, поиск может осуществляться по идентификатору сообщения, статусу сообщения, дате обработки, дате изменения статуса и т. д. В результате поиска пользователю предоставляется технологическая информация из адресной части технологического конверта, XML-представление сообщения, HTML-представление вложенного документа и другие сведения. Также в

программной задаче предусмотрены средства отображения диаграммы последовательности сообщений, сформированных в рамках выбранного рабочего процесса, возможность загрузки сообщения в файл XML-формата для подробного анализа. Для удобства работы с базой данных описания ошибок предусмотрена возможность устанавливать и изменять состояние их рассмотрения: «новая», «в работе» и «исправлено».

В состав обязанностей таможенных органов входит ведение таможенной статистики внешней торговли и специальной таможенной статистики [см. лит.]. В отношении сбора статистики электронного обмена в рабочих процессах таможенных органов нормативными актами регламентируется представление должностными лицами таможенных органов различных видов отчетов с определенной периодичностью. ПАВИС предоставляет пользователям набор регламентированных шаблонов отчетов, который может быть расширен в случае изменений нормативной базы. В текущей версии ПАВИС также существует модуль формирования отчетов на основании существующего набора столбцов. Пользователь выбирает столбцы и располагает их в нужной последовательности. Для удовлетворения требования к периодическому представлению определенных отчетов в ПАВИС реализована функция выполнения запросов по расписанию. На экранной форме пользователю предлагается выбрать шаблон отчета и задать период формирования отчета по заданному шаблону. В результате нужные отчеты формируются автоматически.

Дополнительные возможности диагностики эффективности электронного обмена предоставляются за счет протоколирования работы сервера приложений, круглосуточного мониторинга состояния компонентов автоматизированной системы таможенных органов, в том числе наличия места в табличных пространствах Oracle, на диске, загруженности очередей сообщений. Мониторинг проводится в режиме, близком к реальному времени и позволяет заранее запланировать проведение работ по обслуживанию системы, резервированию данных и запуску наиболее ресурсоемких запросов к базе данных.

При разработке ПАВИС использованы интегрированная среда разработки программного обеспечения IntelliJ IDEA 11.1.3, интегрированная среда разработки на языке SQL Oracle SQL De-

veloper, CASE-средство для проектирования и разработки баз данных ERWin Modeller, среда моделирования программного обеспечения на языке UML Visual Paradigm. Разработка программного кода велась на Java.

В настоящее время система сдана в эксплуатацию, ОАО «НИЦ СПб ЭТУ» проводит ее авторское сопровождение. Результаты опытной эксплуатации показали, что функциональные возможности системы позволяют проводить круглосуточный мониторинг работоспособности модернизированной

автоматизированной системы таможенных органов, анализировать причины нештатных ситуаций и помогают выбирать действия, необходимые для устранения их последствий.

СПИСОК ЛИТЕРАТУРЫ

О таможенном регулировании в Рос. Федерации: Федеральный закон Рос. Федерации от 27.11.2010 № 311-ФЗ // Собр. законодательства Рос. Федерации. 2010. № 48. Ст. 6252.

A. A. Liss, A. S. Skripnikova

AUTOMATED SYSTEM FOR ARCHIVING AND AUDIT OF ELECTRONIC DATA INTERCHANGE IN THE CUSTOMS BUSINESS PROCESSES

The article deals with the automated system developed in the framework of the modernization of the informational and technical Russian Customs support. The processes of archiving and audit of electronic messages, forming in the course of customs activities, are discussed. The possibility of using this system to identify patterns of electronic exchange and tactics of response to emergency situations is underlined.

The automated system of customs authorities, electronic submission, electronic message, information exchange, emergency situation, data recovery

УДК: 20.53.19, 28.23.13

И. И. Холод

Применение методов Data Mining для оценки выполнения программных мероприятий предприятиями ОПК

Описываются подходы применения методов Data Mining для решения различных задач, связанных с оценкой выполнимости программных мероприятий предприятиями оборонно-промышленного комплекса.

Интеллектуальный анализ данных, Data Mining, риски, оценка рисков

Неопределенности финансового, экономического и технологического характера, которые в настоящее время сопутствуют деятельности предприятий оборонно-промышленного комплекса (ОПК), определяют возможности появления соответствующих рисков невыполнения ими мероприятий федеральных целевых программ (ФЦП) и государственного оборонного заказа (ГОЗ). В этих условиях появляется необходимость разработки методов оценки финансово-экономических и технологических рисков, возникающих в процессе создания предприятиями отрасли продукции военного, двойного и гражданского назначений, позволяю-

щих прогнозировать состояние предприятий промышленности и оценивать риски невыполнения ими программных мероприятий.

В настоящее время существуют различные информационные системы для сбора и хранения информации о финансово-экономических параметрах предприятия, его технических возможностях, кадровом потенциале, а также среды его функционирования. В качестве примеров можно привести системы, созданные по заказу Министерства промышленности и торговли РФ:

– автоматизированный реестр организаций ОПК;

– единая информационная система ОПК.

Помимо информационных систем существует базы данных о выполнении предприятиями государственных контрактов:

- единый реестр государственных и муниципальных контрактов;
- реестр организаций;
- реестр недобросовестных поставщиков и др.

Перечисленные и им подобные системы формируют распределенное (поскольку системы расположены в различных ведомствах) информационное пространство, хранящее историю выполнения государственных контрактов. Вся информация может быть однозначно связана по уникальным идентификаторам: ИНН, расчетные счета и др. Объем и разнородность информации в таком хранилище очень велики, в связи с чем применение обычных статистических методов анализа неэффективно как с точки зрения получаемой с их помощью информации, так и с точки зрения времени их работы. Альтернативой могут служить методы интеллектуального анализа данных (в зарубежной литературе эта область имеет название Data Mining).

Методы интеллектуального анализа данных могут быть применены для решения различных задач в области анализа рисков невыполнения мероприятий ФЦП и ГОЗ:

- выявление потенциально недобросовестных поставщиков (исполнителей);
- выявление неисполнимых мероприятий;
- определение параметров отбора исполнителей;
- прогнозирование результатов выполнения предприятиями контракта и др.

Задачи выявления потенциально недобросовестных поставщиков (исполнителей) и неисполнимых мероприятий могут решаться методами классификации на основании информации о ранее не выполненных контрактах и характеристиках их исполнителей. Классическое описание задачи классификации имеет следующий вид [см. лит.]: имеется множество объектов

$$I = \{i_1, i_2, \dots, i_j, \dots, i_n\},$$

где i_j – исследуемый объект.

Таковыми объектами в рассматриваемых задачах являются выполняемые мероприятия и их исполнители.

Каждый объект характеризуется набором переменных:

$$I_j = \{x_1, x_2, \dots, x_h, \dots, x_m, y\},$$

где x_h – независимая переменная, значения которой известны и на их основании определяется значение зависимой переменной y . В задаче выявления потенциально недобросовестных поставщиков независимыми переменными являются финансово-экономические, технологические, кадровые и другие характеристики предприятий ОПК. Примерами таких характеристик могут служить:

- отношение кредиторской/дебиторской задолженностей;
- рентабельность производственных фондов;
- среднесписочная численность работников;
- средний возраст сотрудников;
- отрасль предприятия;
- образование генерального директора;
- средняя заработная плата;
- регион дислокации предприятия и др.

В задаче выявления неисполнимых мероприятий независимыми переменными будут характеристики самих мероприятий (срок, цена, технические характеристики разрабатываемых изделий и т. п.).

В Data Mining набор независимых переменных часто обозначают в виде вектора:

$$X = \{x_1, x_2, \dots, x_h, \dots, x_m\}.$$

Каждая переменная x_h может принимать значения из некоторого множества

$$C_h = \{c_{h1}, c_{h2}, \dots\}.$$

Если значениями переменной являются элементы конечного множества, то говорят, что она имеет категориальный тип. Например, переменная «отрасль предприятия» принимает значения на множестве значений (авиационная, судостроительная, радиоэлектронная...). Независимая переменная может принимать числовые значения в некотором диапазоне (значением переменной «средний возраст сотрудников» может быть целое число в диапазоне от 18 до 70). Возможны типы числовых переменных «без ограничений» или «иметь ограничения снизу или сверху». Например, переменная «среднесписочная численность работников» имеет ограничение снизу и может изменяться в диапазоне от 0 до $+\infty$.

Если множество значений $C = \{c_1, c_2, \dots, c_r, \dots, c_k\}$ зависимой переменной y конечное, то задача является задачей классификации. Например, в задаче выявления недобросовестных поставщиков зависимой переменной является «исполнение контракта», а ее значениями – {выполнен в срок, выполнен с задержкой, не выполнен}. Если пере-

менная y принимает значения на множестве действительных чисел R , то задача называется задачей регрессии. В рассматриваемых задачах такой переменной может быть «срок задержки выполнения», чьи значения меняются в диапазоне от 0 до $+\infty$.

Приведение указанных задач к задаче классификации или регрессии позволяет применить к ним известные методы Data Mining: построения деревьев решений (ID3, C4.5 и др.), аппроксимации (метод наименьших квадратов, SVM и др.), построения классификационных правил (1R, Naïve Bayes и др.). Все перечисленные методы должны применяться к накопленной информации о выполнении контрактов, их исполнителях и значениях показателей на период выполнения контракта.

В результате применения данных методов будут построены классификаторы, позволяющие по параметрам предприятия и программного мероприятия определять добросовестность исполнителя и/или выполнимость данного мероприятия. Полученные классификаторы могут уточняться как применением других методов Data Mining, так и их обучением на вновь поступивших данных.

Задача определения параметров для отбора исполнителей может решаться методами кластеризации на основании анализа характеристик исполнителей, имеющих разные результаты в выполнении контрактов.

Формально задача кластеризации описывается следующим образом [1]. Дано множество объектов данных I , каждый из которых представлен набором атрибутов. Требуется построить множество кластеров C и отображение F множества I на множество C , т. е. $F: I \rightarrow C$. Отображение F задает модель данных, являющуюся решением задачи.

Множество I определим следующим образом:

$$I = \{i_1, i_2, \dots, i_j, \dots, i_n\},$$

где i_j – исследуемый объект.

В задаче определения параметров для отбора исполнителей объектами множества I также будут предприятия, атрибутами – их характеристики. Для определения наиболее значимых атрибутов, опираясь на которые можно выбирать предприятия для выполнения программных мероприятий, необходимо выполнять кластеризацию на множестве объектов I с использованием различных комбинаций атрибутов. Комбинация, которая приведет к разбиению на кластеры наилучшим образом (будет максимально соответствовать группам выполненных и невыполненных контрактов), и будет наиболее значимой.

Кластеризация может осуществляться различными методами Data Mining: построением центроидных кластеров (K-Means, Fuzzy K-Means и др.), иерархическими методами (дивизимными и агломеративными) и др.

Выбирать комбинации можно простым перебором или привлекая эксперта. Первый вариант наиболее прост, может быть автоматизирован, но требует большего времени. Второй вариант может быть выполнен быстрее, но требует привлечения эксперта, который, кроме того, может пропустить комбинацию атрибутов, представляющую интерес.

Данная задача может решаться как для всех видов контрактов сразу, так и для каждого вида в отдельности. Разделение контрактов на виды может осуществляться по разным признакам: объемам финансирования, области разработки и др. В первом случае будут выявлены общие характеристики предприятий, значимые для любого типа контракта. Во втором случае будут определены характеристики, значимые для определенного вида контракта. При использовании полученной информации (о значимости атрибутов) наиболее эффективно будет объединить характеристики, полученные разными способами.

Задача прогнозирования результатов выполнения предприятиями контракта может быть решена методами анализа временных рядов.

Временным рядом называется последовательность событий, упорядоченных по времени их наблюдения. События обычно фиксируются через равные интервалы времени T и представляются в виде последовательности

$$\{e_1, e_2, \dots, e_i, \dots, e_n\},$$

где e_i – событие в момент времени t_i ; n – общее количество событий. В рассматриваемой задаче такими событиями могут быть: сдача этапов контракта, смена руководителей работ, выдача кредитов и другие события в деятельности предприятия.

Задачу построения прогноза по временному ряду можно сформулировать следующим образом: пусть дан временной ряд $\{e_1, e_2, \dots, e_i, \dots, e_n\}$, требуется на его основании определить значение e_{n-k} при $k > 0$. Например, определить, будет ли сдан очередной этап, на основании событий, произошедших в деятельности предприятия в период выполнения этапа.

Прогнозирование временных рядов осуществляется в три этапа:

– построение модели, характеризующей временной ряд. Для этого применяются различные методы классификации;

– оценка построенной модели. Имеющиеся данные разбиваются на два множества: обучающее и тестовое. Построение модели выполняется на обучающем множестве, а затем с ее помощью строят прогноз на тестовом множестве. Спрогнозированные результаты сравнивают с реальными данными и по степени ошибки оценивают модель;

– если построенная на первом этапе модель получила удовлетворительную оценку, то ее можно использовать для прогноза будущих событий.

Данная задача может решаться как методами математической статистики (экстраполяция, экспоненциальное сглаживание и др.), так и методами Data Mining, например методом скользящего окна.

Основная идея метода скользящего окна представлена гипотезой существования некоего закона, по которому можно определить значение очередного члена ряда как функцию от нескольких предыдущих членов. Обычно из каких-то соображений фиксируют число k и предполагают, что только k предшествующих членов влияют на дальнейшее поведение ряда, а зависимость от остальных пренебрегают, т. е.

$$e_{n+1} = f(e_n, e_{n-1}, \dots, e_{n-k}).$$

При этом говорят об «окне» размером k , в пределах которого рассматривается ряд. Для нахождения функции f временной ряд «нарезается» на множество окон (каждое из которых сдвигает-

ся на один элемент). На полученном множестве выполняется поиск искомой функции.

Необходимо заметить, что если функция f используется для предсказания численных значений, то говорят о задаче регрессии. В случае категориальных значений ряда речь идет о классической задаче классификации. Решаются эти задачи методами Data Mining. Например, задача регрессии может быть решена методом SVM, а задача классификации – методами построения деревьев решений. При использовании метода деревьев решений полученные результаты легко представить в виде правил < если–то >. В этом случае в условной части (если) указываются уже прошедшие события, а в заключительной (то) – предсказываемые.

Описанные задачи оценки рисков невыполнения программных мероприятий не ограничиваются рассмотренным списком. Другие задачи в этой области также могут решаться методами Data Mining. Основным требованием для этого является доступность всей необходимой для анализа информации.

СПИСОК ЛИТЕРАТУРЫ

Анализ данных и процессов / А. А. Барсегян, М. С. Куприянов, И. И. Холод, М. Д. Тесс, С. И. Елизаров. СПб.: БХВ-Петербург, 2009.

I. I. Kholod

APPLICATION OF DATA MINING FOR PERFORMANCE EVALUATION PROGRAM ACTIVITIES DEFENSE COMPANIES

The article describes the application of the approaches of Data Mining methods for solving various problems associated with the assessment of the feasibility of the program activities of defense-industrial complex.

Data Mining, risks, risk assessment

УДК: 20.53.19, 28.23.13

И. И. Холод, С. В. Родионов

Построение централизованных хранилищ данных для систем управления предприятиями ОПК

Описывается построение централизованных хранилищ данных для систем поддержки принятия решений, используемых в управлении предприятий оборонно-промышленного комплекса.

Системы поддержки принятия решений, хранилища данных

В настоящее время существует множество информационных систем содержащих различную информацию о деятельности предприятий обо-

ронно-промышленного комплекса (ОПК). Такие системы разрабатываются по заказам различных ведомств, на разных уровнях (отраслевом и феде-

ральном), в разное время, построены на разных технологиях и работают в различных средах. Такое разнообразие обусловлено независимостью формирования требований к этим системам. Несмотря на это целевые задачи у большинства из них схожие: сбор и анализ информации о предприятиях ОПК для эффективного принятия решений по управлению ими как на уровне министерств, так и на отраслевом уровне. В результате во всех них хранится информация об одних и тех же предприятиях. Информация, во многом дополняющая друг друга, а иногда и противоречивая.

В целом вся собираемая информация имеет большой интерес с точки зрения ее анализа для выработки эффективных управленческих решений в развитии как отдельного предприятия ОПК, так и всей отрасли в целом. Для ее использования необходимо обеспечить сбор и хранение актуальной информации о предприятиях ОПК на уровне Министерства промышленности и торговли РФ. Решение данной задачи имеет два уровня:

1) организационный – определяющий нормативную базу, обязывающую предприятия и ведомства предоставлять актуальную информацию;

2) технический – обеспечивающий возможность передачи информации о предприятиях в единое информационное хранилище.

На организационном уровне процесс сбора информации можно разделить на три логических этапа:

- формирование нормативной базы для сбора информации;
- согласование регламента предоставления данных;
- подключение организации к информационному взаимодействию.

На техническом уровне определяются и реализуются технические средства, обеспечивающие сбор актуальной информации. При этом должны решаться следующие вопросы:

- каналы передачи информации;
- форма предоставления информации;
- сценарии передачи информации.

Каналами передачи информации могут быть:

- сети общего пользования – Интернет;
- ведомственные/закрытые сети передачи данных;
- системы передачи данных, предназначенные для передачи секретной информации (например, СПД «Атлас»).

При этом все перечисленные каналы связи могут использоваться совместно в зависимости от

их технической реализации на момент передачи информации, а также от уровня секретности передаваемой информации.

В зависимости от используемых каналов связи будут различаться режимы передачи информации:

- on-line – информация будет запрашиваться модулем загрузки по необходимости посредством вызова web-сервиса (протокола SOAP), в результате на стороне источника информации будет формироваться «обменный файл», который и будет передаваться;

- off-line – передача информации будет осуществляться периодически (регламентировано) посредством выгрузки данных в «обменный файл» на стороне источника информации и его загрузки в единое хранилище.

Реализация обоих вариантов не будет различаться практически ничем, за исключением непосредственно способа передачи «обменного файла», что позволяет применять для этих вариантов одни и те же программные модули.

Использование для передачи информации «обменного файла» позволит передавать информацию непосредственно из информационной системы источника информации. Для этого необходима доработка информационных систем в части формирования «обменного файла» и его передачи в единое хранилище.

Учитывая разнородность информационных систем как с точки зрения платформ, на которых они функционируют, так и с точки зрения технологий на которых они построены; наиболее эффективным представляется использование в качестве формата «обменного файла» формат XML (eXtensible Markup Language).

Уровень организации хранилища данных должен обеспечивать организацию средствами СУБД хранения следующих типов данных (см. рисунок) [см. лит.]:

- детальные данные;
- агрегированные данные;
- метаданные.

Детальные данные должны соответствовать данным, поступающим непосредственно из оперативных источников данных. Они должны извлекаться из отчетов, предоставляемых предприятиями отрасли.

Агрегированные данные должны вычисляться на основании детальных данных и соответствовать наиболее часто запрашиваемой при анализе информации. Архитектура хранилища данных должна предоставлять быстрый и удобный способ получать

информацию, интересующую пользователя. Для этого необходимо часть агрегированных данных хранить в хранилище данных, а не вычислять их при выполнении аналитических запросов.

Метаданные должны обеспечивать пользователя информацией о содержащихся в хранилище данных. Метаданные должны включать в себя описания:

- объектов: метаданные должны описывать объекты предметной области, сведения о которых хранятся в хранилище данных. Такое описание должно включать: атрибуты объектов, их возможные значения, соответствующие поля в информационных структурах хранилищ данных, источники информации об объектах и т. п.;

- пользователей: метаданные должны описывать категории пользователей, использующих данные. Они должны описывать права доступа к данным, а также включать в себя сведения о пользователях, выполнявших над данными различные операции (ввод, редактирование, загрузку, извлечение и т. п.);

правление ошибок, расщепление полей и т. п.), так и в процессе их эксплуатации в хранилище данных;

- времени: метаданные должны описывать время выполнения разных операций над данными (например, загрузка, агрегирование, архивирование, извлечение и т. п.);

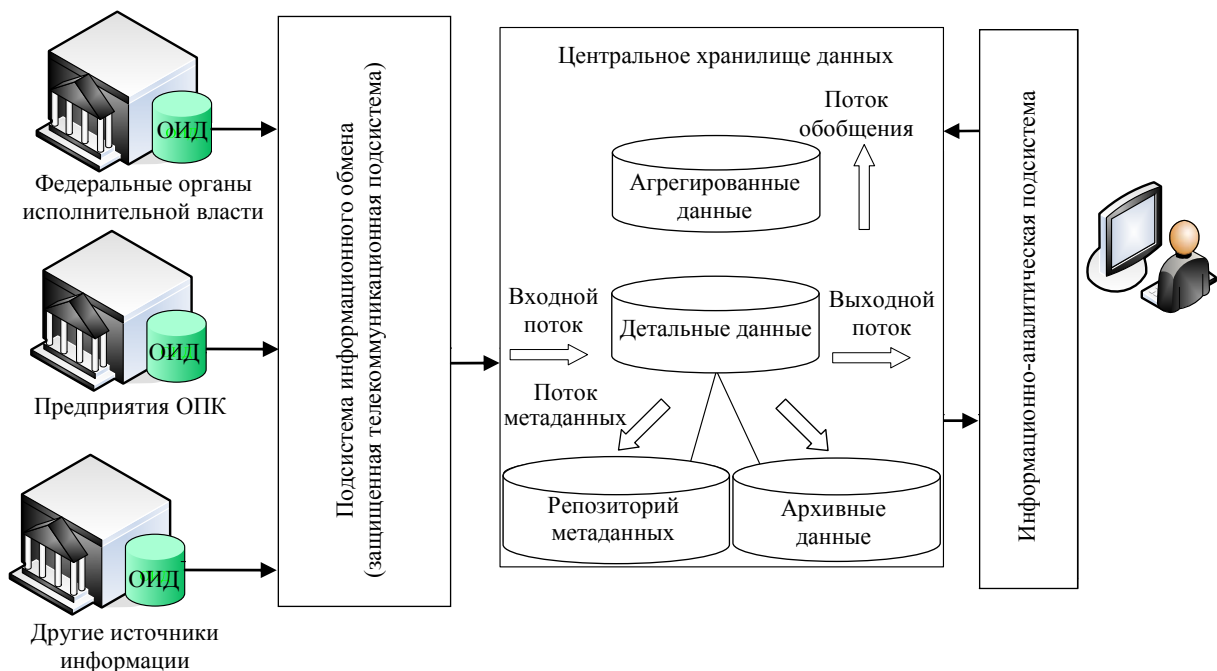
- причин: метаданные должны описывать причины, повлекшие выполнение над данными тех или иных операций. Такими причинами могут быть требования пользователей, статистика обращений к данным и т. п.

Средства центрального хранилища данных (ЦХД) должны обеспечивать циркуляцию следующих потоков информации:

- входной поток: образуется данными, копируемыми из оперативного источника данных (ОИД) в хранилище данных;

- поток обобщения: образуется агрегированием детальных данных и их сохранением в хранилище данных;

- архивный поток: образуется перемещением детальных данных, количество обращений к которым снизилось;



- места хранения: метаданные должны описывать местоположение серверов, рабочих станций, оперативных источников данных, размещенных на них программных средств и распределение между ними данных;

- действий: метаданные должны описывать действия, выполняемые над данными, действия, которые выполняются как в процессе переноса из оперативных источников данных (например, ис-

- поток метаданных: образуется потоком информации о данных в репозиторий данных;

- выходной поток: образуется данными, извлекаемыми пользователями.

Средства центрального хранилища данных должны обеспечивать очистку данных, поступающих из различных источников – федеральных органов исполнительной власти, предприятий

ОПК и других источников. Очистка данных должна обеспечивать снижение вероятности попадания в хранилище ошибочной и/или противоречивой информации, связанной с неправильными преобразованиями форматов, ошибками ввода и т. п.

Реализация центрального хранилища данных, обеспечивающего хранение всей доступной информации по предприятиям ОПК, позволит решать различные задачи как по управлению отдельными предприятиями и отраслями в целом, так и по эффективному размещению государственного обо-

ронного заказа с минимальными рисками невыполнения соответствующих контрактов. Для решения этих задач к созданному хранилищу данных должен быть предоставлен доступ различных аналитических систем, использующих хранимую в нем информацию.

СПИСОК ЛИТЕРАТУРЫ

Методы и модели анализа данных: OLAP и Data Mining / А. А. Барсегян, М. С. Куприянов, В. В. Степаненко, И. И. Холод. СПб.: БХВ-Петербург, 2004. 336 с.

I. I. Kholod, S. V. Rodionov

CREATION OF CENTRALIZED STORAGE FOR DECISION SUPPORT SYSTEMS FOR DEFENSE COMPANIES

This paper describes the construction of a centralized data warehouse for decision support systems used in the management of the defense-industrial complex.

Decision support systems, data warehouses

УДК 681.5.01

С. А. Беляев

Реализация положений ITIL/ITSM в подсистеме управления и контроля функционирования ГАС «Выборы»

Описываются особенности реализации библиотеки инфраструктуры информационных технологий (ITIL/ITSM) в рамках государственной автоматизированной системы «Выборы».

Процессы предоставления услуг, поддержка услуг, информационная автоматизированная система

Успешность применения информационных автоматизированных систем в соответствии с современными подходами определяется тем, насколько они эффективно обеспечивают выполнение бизнес-процессов организации. Применительно к государственной автоматизированной системе (ГАС) «Выборы» это означает, что ее методы и средства должны быть направлены на эффективное и непрерывное обеспечение задач, связанных с подготовкой и проведением избирательных кампаний и референдумов различных уровней [см. лит]. Подсистема управления и контроля функционирования как раз и обеспечивает работоспособность ГАС «Выборы» в целом, а следовательно, решение основной задачи.

Современные тенденции по формированию стандартов, таких как ISO-9000, СММ, СММІ, а также подходов PMBOK и SWEBOOK описывают выполнение требований к организации, проекту или процессу разработки программного обеспечения с точки зрения составляющих процессов. Не является исключением и ITIL (the IT Infrastructure Library – библиотека инфраструктуры информационных технологий), описывающая лучшие из применяемых на практике способы организации работы подразделений или компаний, предоставляющих услуги в области информационных технологий. В настоящее время ITIL становится стандартом для Российского ИТ-рынка, фактически став международным стандартом в

описании фундаментальных процессов управления ИТ-услугами. На ее основе многие компании создали свои структурированные подходы к управлению ИТ-услугами, например HP ITSM Reference Model (HewlettPackard), IT Process Model (IBM), Microsoft Operation Framework (Microsoft). В общем случае данные подходы являются расширением подходов ITIL.

Наиболее известное подмножество библиотеки ITIL – библиотека IT Service Management (ITSM), описывающая процессный подход к предоставлению и поддержке ИТ-услуг. Применение практик ITSM позволяет измерять и контролировать качество услуг ИТ-подразделений и специализированных ИТ-компаний, которые зачастую сталкиваются с недостаточной зрелостью данных процессов.

ITSM рекомендует сосредоточиться на клиенте и его потребностях, на услугах, предоставляемых пользователю информационными технологиями. При этом процессная организация предоставления услуг и наличие эталонных параметров эффективности позволяет ИТ-подразделениям предоставлять качественные услуги, измерять и улучшать их качество. Процессы, описываемые в ITSM, могут быть объединены в две группы: предоставления и поддержки услуг.

Процессы предоставления услуг ITSM:

- управление релизами предназначено для сохранения работоспособности системы при проведении изменений;
- управление уровнем сервиса предназначено для определения необходимого состава и уровня сервисов и инициирования действий по устранению некачественного сервиса;
- управление мощностью предназначено для контроля производительности программно-технических средств и предупреждения инцидентов, вызванных недостаточной мощностью вычислительной инфраструктуры;
- управление доступностью предназначено для определения, измерения и контроля доступности сервисов;
- управление непрерывностью предназначено для обеспечения гарантированного восстановления вычислительной инфраструктуры в случаях нештатных ситуаций.

Процессы поддержки услуг ITSM:

- управление инцидентами предназначено для скорейшего устранения инцидентов, под которыми понимаются любые события, требующие ответной реакции;

– управление проблемами предназначено для выявления и устранения причин инцидентов, предупреждение инцидентов;

– управление конфигурациями предназначено для создания и поддержки в актуальном состоянии логической модели информационной инфраструктуры;

– управление изменениями предназначено для координации и контроля изменений в инфраструктуре;

– служба Service Desk предназначена для управления обращениями пользователей.

С точки зрения ITIL предоставление услуг должно обладать ценностью, которая состоит из полезности, включающей в себя обеспечение производительности или снятие ограничений, и гарантии выполнения, включающей в себя доступность, мощность, непрерывность и безопасность*. Перечисленные процессы ITSM уточняют положения ITIL.

Практики ITSM описывают основные принципы, которые должны быть реализованы, но вариаций реализации может быть множество, в частности, для поддержки работы службы Service Desk может использоваться как специализированное программное средство, так и решение известного вендора или обычная электронная таблица. При этом ITSM не ограничивает возможных реализаций.

Исходя из рекомендаций библиотеки ITSM и потребностей ГАС «Выборы» приняты следующие подходы к реализации подсистемы управления и контроля функционирования.

- Управление релизами. ГАС «Выборы» – активно развивающаяся система, причиной тому служит постоянно изменяющееся законодательство, в связи с чем программное обеспечение обновляется на регулярной основе. Для контроля релизов реализована автоматическая инвентаризация специального программного обеспечения и структуры базы данных всех избирательных комиссий Российской Федерации.

- Управление уровнем сервиса. ГАС «Выборы» – географически распределенная информационная система, в которой доступ к большей части информационных сервисов осуществляется удаленно. Для контроля уровня сервиса и управления им предусмотрена система мониторинга функционирования локальной вычислительной сети,

* ITIL. IT Service Management по стандартам v.3.1. URL: // <http://www.intuit.ru/studies/courses/2323/623/lecture/8120>.

распределенной по всей территории Российской Федерации, с возможностью информирования при возникновении нарушений в ее работе.

- **Управление мощностью.** Основным ресурсом при предоставлении сервисов в ГАС «Выборы» являются серверы, расположенные в Федеральном центре информатизации при Центральной избирательной комиссии и в информационных центрах субъектов Российской Федерации. Для контроля уровня сервиса предусмотрен мониторинг производительности серверов и средства отображения данной производительности на интерактивной карте, позволяющие не только локализовать места перегрузки, но и проанализировать возможные ее причины.

- **Управление доступностью.** В рамках ГАС «Выборы» используются самые различные каналы связи: оптоволоконные сети, витая пара, ADSL и спутниковая связь. В настоящее время не предусмотрено автоматическое управление доступностью при появлении сбоев в каналах связей, но осуществляется мониторинг доступности сервисов по всей территории Российской Федерации и при наличии сбоев осуществляется автоматическое информирование администраторов, ответственных за соответствующий участок вычислительной сети; автоматически регистрируются также и нештатные ситуации.

- **Управление непрерывностью.** Как в любой большой информационной системе в ГАС «Выборы» большое количество технических средств, которые периодически дают сбои или отказы в работе. Для обеспечения непрерывности работы системы реализованы средства контроля за регламентным обслуживанием технических средств, их ремонтом, заменой и списанием.

- **Управление инцидентами.** В качестве инцидента в ГАС «Выборы» может выступать не только технический, но и программный сбой. Предусмотрены средства автоматического регистрирования данных сбоев в электронном журнале, формирование рекомендаций по устранению на основании базы знаний, а затем контроль их устранения.

- **Управление проблемами.** Под проблемами в ГАС «Выборы» понимаются как инциденты функционирования, так и инциденты, связанные с другими процессами ITSM. Для предупреждения возникновения проблем разработчики подсистем ведут базу знаний, в которой описаны типовые проблемы и методы их решения.

- **Управление конфигурациями.** Информационная инфраструктура ГАС «Выборы» включает в себя более десятки тысяч компьютеров, сетевого

оборудования, которые должны иметь определенную конфигурацию аппаратного и программного обеспечения. Для организации процесса управления конфигурациями разработаны конфигураторы, которые в автоматическом режиме проверяют соответствие установленного и требуемого оборудования и программного обеспечения.

- **Управление изменениями.** Изменения в ГАС «Выборы» возникают как в составе специального программного обеспечения, так и в конфигурации и составе технических средств. Для организации процесса управления изменениями реализованы средства, обеспечивающие автоматическую установку обновлений специального программного обеспечения, а также планирование и контроль поставки, движения, ремонтов и списания технических средств.

- **Служба Service Desk.** Пользователи ГАС «Выборы» делятся на две большие категории: Центральная избирательная комиссия, которая находится в Москве, и избирательные комиссии субъектов Российской Федерации. Для обеспечения единой точки входа по обращениям пользователей реализован Интранет-портал, предназначенный для ввода обращений и контроля ответов на них.

Реализация процессов ITSM в ГАС «Выборы» позволяет говорить о внедрении лучших практик процессной организации управления услугами, однако как и большинство стандартов и практик управления, практики ITSM предполагают непрерывное улучшение услуг. Это связано не столько с возможной неудачной первоначальной реализацией, сколько с изменяющимися условиями, в которых работает система. Для решения задачи непрерывного улучшения Федеральный центр информатизации при Центральной избирательной комиссии на ежегодной основе проводит конкурсы по сопровождению подсистем ГАС «Выборы», благодаря чему обеспечивается не только поддержание их в работоспособном состоянии, но и адаптация к изменяющемуся региональному и федеральному законодательству, устареванию и обновлению парка технических средств.

Подсистемы ГАС «Выборы» ориентированы на проведение избирательных кампаний и референдумов, при этом подсистема управления и контроля функционирования носит универсальный характер, поскольку основной объект ее работы – это программно-технические средства. С точки зрения решаемых задач она может применяться для мониторинга работы большинства современ-

ных географически распределенных систем, например в рамках системы экстренного реагирования при авариях «ЭРА-ГЛОНАСС» или в рамках автоматизированной системы Судебного департамента. Внедрение ITSM в рамках данных систем может оказаться сложной задачей*, что обусловлено необходимостью не только установки и использования программных средств, обеспечивающих автоматизацию всех базовых процессов, но и необходимостью изменения организационных процессов на предприятии, внедряющем ITSM.

Опыт Федерального центра информатизации при Центральной избирательной комиссии Рос-

сийской Федерации говорит о возможности успешной реализации практик ITIL/ITSM в Российской организации с использованием универсальной программной подсистемы, которая может быть повторно использована в других аналогичных организациях.

СПИСОК ЛИТЕРАТУРЫ

Развитие государственной автоматизированной системы Российской Федерации «Выборы» до 2012 г.: Пояснительная записка к техн. проекту. Ч. 10. Подсистема управления и контроля функционирования, ИРЦВ.42 5100 5.001.П2-10.

S. A. Belyaev

IMPLEMENTATION OF ITIL/ITSM IN GAS «ELECTION»' S SUBSYSTEM OF MANAGEMENT AND CONTROL

The specific of implementation of ITIL/ITSM in government automation system «Election» is described in the article.

Servicing processes, services maintain, informational automation system

УДК 005.8

С. А. Романенко

Внедрение системы проектного управления как инструмент повышения эффективности деятельности современной IT-компании

Рассматриваются предпосылки, особенности, проблемы и подходы к внедрению системы проектного управления в современной IT-компании, проводится анализ влияния организационной структуры и используемых методов управления проектами на эффективность деятельности компании, предлагаются критерии оценки успешности внедрения проектного управления.

Проектное управление, организационная структура предприятия, эффективность деятельности, проектный офис

В современных условиях рыночной экономики эффективность деятельности IT-компании определяет ее конкурентоспособность и дальнейшее устойчивое развитие. Эффективность деятельности – экономический показатель, соизмеряющий полученный эффект с затратами или ресурсами, использованными для достижения этого эффекта

[1]. Таким образом, экономическая эффективность отражает связь между результатом деятельности компании и затратами, произведенными для получения этого результата.

Основным видом деятельности большинства IT-компаний является заказная разработка, выполнение проектов. Чем более динамично развивается компания, тем больше проектов выполняются одновременно. При этом проекты различаются по срокам, сложности, прикладной области, применя-

* Простота и сложность ITSM. URL: // <http://www.interface.ru/home.asp?artId=2303>.

емым технологиям, ресурсам, требуемым для их реализации. В условиях сложно-прогнозируемых изменений внешней среды залогом стабильного существования и развития ИТ-компании является диверсификация портфеля заказов, готовность к быстрой адаптации к новым заказчикам, расширению спектра предметных областей и технологий разработки.

Особенностью современного состояния ИТ-отрасли является острый дефицит квалифицированных профессиональных кадров. «Война за таланты» развивается стремительно, что делает задачи эффективного использования специалистов и сокращения показателей текучести персонала первостепенными для менеджмента любой ИТ-компании. Характерные для России исторически сложившиеся в XX в. линейно-функциональные и дивизионные организационные структуры компаний в сочетании с иерархическим типом управления становятся неэффективными в условиях быстроменяющейся окружающей среды, особенно в периоды быстрого роста компании, расширения или изменения структуры портфеля заказов, диверсификации направлений деятельности. Это обусловлено несколькими причинами.

Во-первых, между отдельными частями организаций такого рода возникает не сотрудничество, а конкуренция. Направления получают автономную функциональную структуру, которая позволяет им частично или полностью взять на себя ответственность за разработку, производство и сбыт продукции по определенному направлению деятельности компании или для определенного заказчика. Каждое подразделение, реализуя определенное направление деятельности и независимо развивая необходимые компетенции, фактически становится самостоятельной бизнес единицей внутри предприятия. Если направления деятельности компании технологически близки, то происходит неизбежное дублирование функций (компетенций) внутри подразделений. Это приводит к увеличению затрат вследствие дублирования одних и тех же видов работ для разных видов продукции. В каждом подразделении создаются свои функциональные подразделения, что приводит к неэффективному использованию однотипных ресурсов, особенно в периоды неустойчивого положения одних направлений деятельности и пиковых нагрузок по другим направлениям деятельности компании.

Во-вторых, велика вероятность несовпадения, а часто и противоречия интересов самостоятельных подразделений и общих показателей эффективности деятельности компании. Руководители подразделений в меньшей степени заинтересованы в достижении показателей деятельности компании, чем в развитии направления деятельности, которое они возглавляют. Оценка отдельных сотрудников осуществляется на основе показателей результативности деятельности подразделения в целом, а не личного вклада каждого специалиста, его профессионального уровня. Зачастую оценка носит субъективный характер, так как производится руководителем направления, не являющимся экспертом в конкретных профессиональных компетенциях.

В-третьих, развитие сотрудников возможно только в рамках определенного направления деятельности компании, которое зачастую ограничивает рост профессиональных компетенций и навыков, возможности использования современных технологий и средств разработки.

В-четвертых, линейно-функциональная или дивизионная организационная структура способствуют созданию организаций, сопротивляющихся изменениям, особенно структурным изменениям, поэтому они перерождаются в бюрократические структуры, не поддающиеся адаптации. Существующая обстановка требует, чтобы организации были не только готовы к любым изменениям, но и способны им подвергаться.

В-пятых, линейно-функциональные и дивизионные организационные структуры ограничивают число и характер возможных вариантов оперативного решения возникающих проблем. Становятся невозможны решения, обеспечивающие развитие компании с учетом технических и социальных изменений, темпы которых все больше и больше растут. Остаются общие для иерархических структур недостатки – волокита, перегруженность управленцев, плохое взаимодействие при решении вопросов, смежных для подразделений.

Очевидно, что для преодоления указанных факторов современная ИТ-компания должна иметь достаточно гибкую, проектно-ориентированную, многомерную организационную структуру. При этом проблемы управления проектами становятся приоритетными и определяющими, с точки зрения эффективности текущей деятельности и раз-

вития IT-компаний. При отсутствии формализованной системы управления неизбежны проблемы, связанные с конфликтами целей, приоритетов, сроков, назначений ресурсов. Успешная реализация проектов в таких условиях, как правило, достигается за счет усилий отдельных менеджеров, а положительный опыт выполнения проектов редко удается использовать в будущем. Именно система управления проектами, основанная на использовании единых принципов и правил, позволяет грамотно планировать и успешно реализовывать большое количество проектов, оптимизируя затраты временных, денежных и человеческих ресурсов, но при этом не отклоняясь от запланированного качества конечного продукта проекта. Использование современных принципов и методов управления проектами является основой для достижения организацией новых конкурентных преимуществ и повышения результативности своей деятельности. Компании, стремящиеся остаться конкурентоспособными, вынуждены постоянно совершенствовать процессы управления, иницируя и проводя в жизнь программы оптимизации методов и технологий управления. Применение проектных подходов в деятельности IT-компаний делает ее более гибкой и адаптивной.

В мире существуют различные стандарты по управлению проектами. Стандарт РМВОК [2], разработанный Институтом управления проектами (Project Management Institute – PMI), является одним из самых распространенных в России и США. Основная цель РМВОК – систематизация знаний по управлению проектами, которые обычно считаются хорошей практикой. «Обычно считается» означает, что описываемые знания и практики применимы к большинству проектов в большинстве случаев, причем относительно их значения и пользы существует консенсус профессионального сообщества. «Хорошая практика» означает, что правильное применение этих навыков, инструментов и методов способно повысить вероятность успеха для широкого круга различных проектов.

В соответствии с РМВОК проект – это временное предприятие, предназначенное для создания уникальных продуктов, услуг или результатов. Временный характер означает, что у любого проекта есть начало и завершение. Завершение наступает тогда, когда достигнуты цели проекта,

или признано, что цели проекта не будут или не могут быть достигнуты, или исчезла необходимость в проекте. В этом контексте внедрение системы проектного управления – проект организационного развития, системное изменение, которое трансформирует организационную структуру компании, упорядочивает бизнес-процессы, изменяет людей компании и ее культуру. Успешность такого проекта определяется несколькими факторами, основными из которых являются:

1. *Неудовлетворенность текущим состоянием дел.* Изменения проходят достаточно быстро и безболезненно в том случае, если существует неудовлетворенность не только первых лиц компании, но и всего персонала. Хотя возможна и другая ситуация, когда неудовлетворенность высшего звена очень велика, при том что остальная часть компании не испытывает неудовлетворенности существующим положением дел. Неудовлетворенность лидеров имеет больший вес, чем нейтральная позиция или сопротивление сотрудников, но изменения должны «запускаться» только тогда, когда суммарная неудовлетворенность будет выше, чем суммарное сопротивление изменениям. Важно, чтобы топ-менеджмент смог объяснить невозможность для компании «жить по-старому».

2. *Определенность целей изменений.* Цели должны быть такими, чтобы их можно было представить, реализовать, а их достижение было понятным и желаемым для большинства, в первую очередь для ключевых сотрудников компании. Сформулированные цели изменений должны определить, есть ли реальная потребность во внедрении системы проектного управления.

3. *Наличие образа будущего.* Он должен быть сформирован руководством компании и определять, что будет представлять собой компания после внедрения системы проектного управления. Будущее состояние компании должно быть описано целостно с проработкой деталей со степенью, достаточной для его представления всем сотрудникам компании. Необходимо найти место в этом желаемом будущем для всех участников изменений. Сотрудники перед переменами почти всегда озабочены своим будущим положением в большей степени, чем перспективами развития предприятия. В этих условиях руководству компании важно найти правильные слова, снизить степень неопределенности относительно будущего каждого сотрудника, обеспечить всех вовлеченных в проект людей достоверной информацией, обозначить возможные выгоды для сотрудни-

ков. Именно от руководства компании должен исходить энергетический импульс. Руководство компании не должно снимать с себя ответственность за формирование положительных эмоций в будущем изменении и обязано демонстрировать «правильные эмоции», обеспечивая вовлеченность персонала в изменения. Не менее важен вопрос о том, что произойдет, если проект изменений закончится провалом.

4. *Конкретный план достижения целей.* При составлении плана необходимо рассмотреть несколько альтернативных путей внедрения системы проектного управления (например, различные темпы внедрения, альтернативы организационной структуры, возможные программные продукты управления проектами и т. д.). Использование методов совместного принятия решений при обсуждении альтернативных путей внедрения системы проектного управления и ее содержания снижает вероятность принятия ошибочного решения, а также позволяет вовлечь персонал в процесс принятия решения о способах достижения целей.

5. *Достоверная оценка сопротивления.* Перед началом системных изменений необходимо адекватно определить степень общего сопротивления организации процессу внедрения системы проектного управления. Используя переговоры, необходимо создать коалицию из менеджеров, способных оказывать поддержку изменениям. Если уровень сопротивления велик, надо быть уверенным, что он может быть снижен в процессе изменений. Особое внимание уделяется тому, как члены управленческой команды относятся к участию в проекте изменений. Менеджеры должны первыми пройти обучение. В рамках стратегии внедрения системы проектного управления обучение высшего руководства компании символизирует личное участие руководителей во внедрении системы и служит примером для остальных участников проекта. В программах обучения должны быть задействованы все сотрудники, которым предстоит участвовать в процессах управления проектами. При этом следует учитывать потребность в обучении и составлять гибкие программы обучения, различающиеся по содержательному наполнению и продолжительности.

Дополнительными факторами, влияющими на успех проекта внедрения системы проектного управления, являются:

- результативность и эффективность реализации компанией своих недавних внутренних проектов;

- наличие «усталости от изменений», которая характеризуется отсутствием готовности менеджмента компании и сотрудников к участию в процессах изменения, разочарованием неудачной реализацией предыдущих проектов;

- недопустимость одновременного параллельного запуска нескольких проектов изменения;

- неизменность команды, реализующей изменения на протяжении всего периода внедрения системы проектного управления.

Целями внедрения системы проектного управления обычно являются:

1. Повышение прозрачности, управляемости и прогнозируемости результатов компании при ведении нескольких проектов любой сложности.

2. Повышение доли успешно реализованных проектов, выполненных в установленные сроки в пределах выделенного бюджета и с требуемым качеством продукции.

3. Сокращение издержек на выполнение проектов и потерь, связанных с оплатой простоев производственного персонала, не занятого в проектах.

4. Создание эффективной системы мотивации персонала, обеспечивающей повышение производительности труда, справедливое распределение вознаграждения и удовлетворенность менеджеров и сотрудников проектных команд.

Система проектного управления в общем случае включает в себя организационную и процедурную составляющие. Элементами организационной структуры являются органы управления и звенья организационно-штатной структуры всех уровней, а также взаимосвязи между ними. Процедуры управления проектами определяют порядок, регламенты выполнения участниками проекта своих функций, настроенные на эффективную реализацию работ по проекту.

Переход к проектно-ориентированной компании и проектному управлению в большинстве случаев неизбежен без изменения ее организационно-штатной структуры. Центральными звеньями проектно-ориентированной организационной структуры компании являются офис управления проектами и менеджер проекта.

Офис управления проектами (центр управления проектами, проектный офис) – это подразделение, осуществляющее различные функции, относящиеся к централизации и координации управления проектами, входящими в его сферу ответственности. Сфера ответственности офиса управления проектами может варьироваться от ока-

зания поддержки в управлении проектами до прямого управления проектом. Проекты, поддерживаемые или управляемые проектным офисом, могут быть не связанными, но управляться в совокупности. Конкретная форма, функции и структура офиса управления проектами зависят от потребностей компании, в рамках которой он функционирует. Конкретный проектный офис может получить полномочия действовать как неотъемлемая заинтересованная сторона проектов, имеющая решающее слово в начальной стадии каждого проекта. Он может иметь полномочия давать рекомендации или останавливать проекты, или выполнять другие действия, чтобы цели компании оставались согласованными и непротиворечивыми. Кроме того, проектный офис может участвовать в отборе общих или выделенных ресурсов проекта, управлении ими или их распределении. Основная функция проектного офиса заключается в поддержке управления проектами различными способами, среди которых выделяются следующие:

- управление общими ресурсами всех проектов, администрируемых проектным офисом;
- определение и разработка методологии, лучших практик и стандартов управления проектами;
- коучинг, наставничество, обучение и надзор;
- мониторинг соответствия стандартам, процедурам и шаблонам управления проектами посредством аудитов проектов;
- разработка и управление принципами, процедурами, шаблонами проекта и другой общей документацией (активами процессов организации);
- координация коммуникаций между проектами.

Менеджер (руководитель) проекта несет полную персональную ответственность за результаты выполнения конкретного проекта. Он отвечает за взаимодействие с заказчиком и потребителями продукции, управление бюджетом проекта, качество получаемых результатов, ведение проектной документации, обеспечивает выполнение запланированных работ, готовит предложения по изменениям в планах, координирует процесс использования технических и человеческих ресурсов, задействованных в конкретном проекте. В непосредственное управление руководителю проекта выделяется проектная команда, составленная из специалистов требуемой квалификации, обучен-

ных применению унифицированных технологий и средств разработки, принятых в компании.

Менеджеры проектов и проектный офис преследуют разные цели и, таким образом, руководствуются различными требованиями. Тем не менее, все их действия ориентированы на стратегические интересы организации. Разница между ролью менеджера проекта и проектного офиса может заключаться в следующем:

- менеджер проекта сосредоточивается на конкретных целях проекта, в то время как проектный офис управляет основными изменениями в содержании совокупности проектов (по направлениям деятельности) и может рассматривать их как потенциальные возможности для более успешного достижения целей;
- менеджер проекта управляет ресурсами, переданными проекту, с целью более точного выполнения целей проекта, а проектный офис оптимизирует использование общих ресурсов организации во всех проектах;
- менеджер проекта управляет ограничениями (содержанием, расписанием, стоимостью и качеством и т. д.) отдельных проектов, а проектный офис управляет методологиями, стандартами, общими рисками, возможностями и взаимозависимостями проектов на уровне предприятия.

Организационно-штатная структура проектной ориентированной компании помимо офиса управления проектами включает в себя организационные структуры, направленные на создание условий для развития кадрового и технологического потенциала предприятия. Для этих целей в организационно-штатной структуре создаются центры компетенций, в рамках которых решаются следующие основные задачи:

1. Унификация технологий и средств проектирования и разработки.
2. Накопление, фиксация в базах знаний и контроль использования в проектах интеллектуального актива компании, проверенных технических и технологических решений, алгоритмов, компонентов, библиотек и т. д.
3. Структурная инкапсуляция сотрудников с одинаковыми профессиональными компетенциями и эффективное их распределение по проектным командам, обучение, повышение квалификации персонала с учетом текущих и перспективных потребностей компании.

4. Коллективная выработка оптимальных технических решений по проектам с привлечением наиболее квалифицированных специалистов в конкретной области.

5. Экспертиза и контроль реализации проектов по направлению центра компетенции.

При разработке процедур управления проектами используются как общие подходы, применимые для любой организационной системы, так и частные, учитывающие уникальность компании и выполняемых проектов. Общие подходы обычно используются для «постоянной» составляющей проекта, которая в большей степени определяется его окружением. Частные процедуры ориентированы на управление изменениями, обеспечивают разработку и адаптацию механизмов управления содержанием, качеством, временем, затратами, рисками, трудовыми ресурсами, контрактами и закупками, коммуникацией, всем тем, что способствует своевременному завершению проекта при оптимальных затратах. Процедуры управления проектами формализуются в виде набора нормативных документов (регламентов), которые включаются в систему менеджмента качества компании. К типовым регламентирующим документам системы управления проектами относятся:

1. *Процедура управления проектами* – определяет механизмы отбора, оценки, инициации, управления, выполнения, контроля и завершения проекта, распределение полномочий по проекту, критерии оценки успешности проекта. Процедура управления проектом включает иерархическую декомпозицию управленческих функций в соответствии с решаемыми задачами, описание процедур принятия решений и последовательности решения задач по проекту, механизмы взаимодействия руководителя проекта с руководителями центров компетенции и другими подразделениями компании. Необходим единый контур управления, т. е. решения на разных уровнях управления и между отдельными проектами должны приниматься таким образом, чтобы реализация управленческих решений в рамках отдельного проекта не оказывала негативного влияния на осуществление других проектов или операционную деятельность компании.

2. *Концепция бюджетирования проекта* – определяет порядок разработки, согласования, утверждения, исполнения и контроля фактически-

го расходования средств, выделяемых менеджеру проекта для его выполнения.

3. *Система мотивации проектных команд* – определяет порядок оплаты труда и бонусирования менеджера проекта и членов проектных команд по результатам выполнения проекта. Проектно-ориентированная система мотивации предполагает оплату в рамках выделенного бюджета проекта только того объема работы каждого сотрудника, который дал значимый вклад в общий результат проекта и принят менеджером проекта. «Простои» сотрудников, невостребованных в проектах, а также неквалифицированно выполненная работа сотрудников не должны оплачиваться. Материальная мотивация руководителя проекта и членов проектной команды привязывается непосредственно к плановым и фактическим результатам выполнения проекта по бюджету, срокам и качеству выпускаемой продукции.

4. *Шаблоны типовых проектов* – формализуют основные этапы и контрольные точки выполнения проектов с учетом их специфики и ограничений, накладываемых руководящими документами. Наличие типовых шаблонов для различных типов проектов как инструмента работы руководителя проекта позволяет сократить время запуска проекта, более точно спрогнозировать и смоделировать варианты развития проекта в зависимости от изменения внешних и внутренних условий, повышает качество планирования и выбора наиболее подходящей методологии управления конкретным проектом, увеличивает вероятность успешного выполнения проекта.

5. *Альбом (библиотека) форм и шаблонов документов*, являющихся обязательными для разработки и актуализации в процессе управления проектом и его выполнения. Наличие и использование готовых шаблонов документов повышает прозрачность процесса управления проектом, способствует снижению накладных расходов на решение организационных вопросов, обеспечивает унификацию представления данных о проекте. Единый стандарт представления данных, общие принципы декомпозиции проектов, единый подход к отражению организационной, технологической, информационной и других структур проектов позволяют построить и непрерывно развивать внутрифирменный стандарт управления проектами. К типовым документам управления проектами в соответствии с РМВОК относятся: устав проекта, ре-

есть заинтересованных лиц, иерархическая структура работ, бюджет проекта, план выполнения проекта, реестр и план управления рисками, план закупок, план обеспечения качества, стратегия внешних и внутренних коммуникаций, ретроспективный отчет по проекту.

Важным фактором успеха внедрения системы проектного управления является использование специализированных программных средств для создания единого информационного пространства, обеспечивающего повышение эффективности работы менеджера проекта и возможность получения в любой момент времени достоверной и актуальной информации о проекте. Используемые информационные системы должны быть достаточно гибкими и настраиваемыми, чтобы учесть все существенные особенности управления как самой компанией, так и ее проектами. Программные средства управления отдельными проектами и информационной поддержки проектных команд должны интегрироваться с общей системой управления компанией. Современные информационные системы обеспечивают возможность сбора и анализа информации о фактическом выполнении планов проекта, укомплектованности проектных команд, расходовании бюджета, выполнении мероприятий обеспечения качества продукции и т. д. На основании результатов анализа на ранних стадиях выполнения проекта могут быть выявлены отклонения, которые, в случае несвоевременного реагирования на них, приведут к неудаче проекта в целом.

Для оценки результатов внедрения системы проектного управления, как и для любого другого проекта, необходимы критерии успешности, которые отражают изменения качественных и количественных показателей эффективности деятельности компании. Критерии успешности внедрения системы проектного управления для каждой компании специфичны. Однако можно сформулировать типовые показатели, которые применимы в большинстве случаев для сравнения эффективности деятельности компании до и после внедрения системы проектного управления:

1. Доля проектов, выполненных в установленные (плановые) сроки, в рамках выделенного бюджета и с качеством, требуемым техническим заданием на проект.

2. Выработка производственного персонала – показатель, определяющий финансовый показатель объема выполненной работы (например, собственный объем работ в денежном выражении) на единицу произведенных затрат (единице оплаты труда производственного персонала).

3. Показатель «утилизации» производственного персонала, вычисляемый как отношение общего фонда заработной платы сотрудников, занятых в проектах, к общему фонду заработной платы производственного персонала компании.

4. Укомплектованность проектных команд в моменты запуска, выполнения и завершения проектов.

5. Отклонения по стоимости проекта – отклонения бюджета проекта, вызванные его перерасходом или недорасходом.

6. Отклонения в расписании – сдвиги в расписании проекта, вызванные отставанием или опережением планового графика работ.

Отклонения по качеству – отношение затрат ресурсов (денежных, человеческих, временных), использованных для устранения дефектов и недостатков, найденных при проверке качества продукции, к общему объему затрат, понесенных на проектирование, разработку и выпуск продукции.

Сегодня, чтобы преуспеть в конкурентной борьбе, необходимо обеспечить оптимальное сочетание отлаженных бизнес-процессов в структурах управления с динамичными и нацеленными на конечный результат проектными подходами. Повышение эффективности деятельности современной IT-компании возможно за счет систематизации проектной деятельности (внедрения систем управления проектами), использования современных инструментов и методов планирования и контроля проектов, применения знаний и мирового опыта проектного управления.

СПИСОК ЛИТЕРАТУРЫ

1. Анализ эффективности деятельности предприятия: Учеб. пособие / И. И. Мазурова, Н. П. Белозерова, Т. М. Леонова, М. М. Подшивалова. СПб.: Изд-во СПбГУЭФ, 2010. 113 с.

2. Руководство к своду знаний по управлению проектами (Руководство РМВОК). 4-е изд. / Project Management Institute, Inc., 2008. 496 с.

S. A. Romanenko

INTRODUCTION OF PROJECT MANAGEMENT SYSTEM AS TOOL OF ACTIVITY EFFICIENCY INCREASING OF MODERN IT-COMPANY

The conditions, features, problems and methods of introduction of project management system in modern IT-company are described in the article. Analysis of influence of organizational structure and project management techniques on company efficiency, criteria of introduction successfulness are also described in the article.

Project management, company's organizational structure, activity's efficiency, project management office

УДК 681.518.5+681.5

М. Ю. Шестопапов

Системный подход к построению математических моделей объектов отказоустойчивого управления

В качестве объектов отказоустойчивого управления рассматриваются системы управления технологическими процессами, представляемые как множества подсистем со своими функциями и целями по получению и обработке информации, между которыми происходит передача информации по ориентированным каналам связи.

Объекты отказоустойчивого управления, управление технологическими процессами

Построение математических моделей объектов отказоустойчивого управления представляет собой сложную научно-техническую проблему, связанную с необходимостью комплексного решения как диагностики состояния технологических систем и объектов, так и собственно обеспечения отказоустойчивости управления, и должно основываться на едином, системном подходе. Особенность системного подхода состоит в том, что система исследуется как единый организм с учетом связей между различными элементами и внешних связей с другими системами, а управление – как процесс, обеспечивающий требуемое поведение системы. Данная предметная область требует применения нового методологического подхода, опирающегося на принципы системологии. Следовательно, методология теории систем, системного анализа и теории управления, хорошо развитая применительно к причинно-следственным моделям, вполне применима и к объектам отказоустойчивости и их системам диагностики [1]–[3].

Системы отказоустойчивого управления (СОУ) техническими системами являются информационно-

алгоритмическими комплексами. К их моделированию следует применить принципы системного подхода к построению моделей, анализу и эволюционному синтезу сложных систем управления [2]–[4]. Сложная система управления представляет собой множество взаимосвязанных и взаимодействующих подсистем, выполняющих собственные и общесистемные функции и цели [3], [4].

Глубина диагностирования, точность локализации неисправностей и возможность восстановления качества основных процессов напрямую зависят от объема априорной информации об устройстве объектов отказоустойчивого управления (ООУ). Для построения моделей с раскрытой структурой систему расчленяют на подсистемы и элементы с учетом потенциальных неисправностей и планируемой глубины их диагностирования. Для каждой части строится своя математическая модель, а затем модели частей связывают между собой таким же образом, как соединялись сами части.

В работе предлагается рассматривать причинно-следственные модели ООУ, которые включают:

- описание номинальной системы с раскрытой структурой;
- описание среды как источника сигнальных и системных неисправностей;
- перечисление переменных систем управления технологическим процессом (СУ ТП), которые допускают непосредственное измерение;
- перечисление подстраиваемых параметров, реструктурируемых операторов подсистем и реконфигурируемых связей.

Описание взаимодействия ООУ со средой начинается с выделения собственно системы S и ее связей со средой на входе F и выходе Y . Среда ООУ разделяется на среду как источник неисправностей F_F (возмущения) и среду как меры по восстановлению F_U (управляющие воздействия) (рис. 1).

Объектами теоретических исследований СОУ являются модели собственно системы M_S ; M_{YS} , M_{SF} – модели взаимодействия с внешней средой

$$M_{YSF} = \langle M_{YS}, M_S, M_{SF} \rangle$$

и модели расширенных систем [4], [5]

$$M_{RS} = \langle M_{YSF}, M_S \rangle.$$

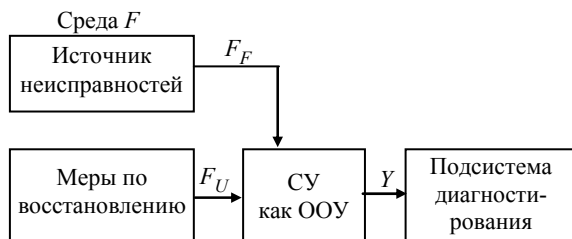


Рис. 1

Для описания связей системы со средой M_{YSF} модель собственно системы M_S дополняется указанием звена (вершины сигнального графа), на вход которого подается воздействие, и звена, выход которого является выходом системы. В результате задания структур генераторов, моделирующих номинальные сигнальные воздействия среды, получают модели M_{RS} .

Переход от моделей собственно системы M_S к системе со связями M_{YSF} означает дополнительную информацию о связях системы со средой на выходе M_{YS} и на входе M_{SF} . Переход от системы со связями M_{YSF} к модели расширенной системы M_{RS} – раскрытие неопределенности о среде.

Модели M_S описывают свободные движения системы, т. е. свойства устойчивости и качества процессов при ненулевых начальных условиях. Модели M_{YSF} описывают каналы передач от входов к выходам системы, т. е. свойства инвариантности к любым возмущениям, а модели M_{RS} привлекаются для изучения вынужденных движений – селективной инвариантности – независимости (слабой зависимости) установившихся составляющих движений к конкретным типам возмущений [6].

Интерпретация процесса построения моделей систем как последовательного раскрытия неопределенности позволяет сформулировать следующие утверждения:

Утверждение 1. Свойства, присущие менее определенной модели, сохраняют силу при повышении информативности модели.

Доказательство следует из того, что менее определенные модели представляют собой множества более определенных моделей, что можно записать так:

$$M_S = \{M_{YSF}\}; M_{YSF} = \{M_{RS}\}.$$

Следовательно, свойства M_S относятся ко всему множеству $\{M_{YSF}\}$, а свойства M_{YSF} – ко всему множеству $\{M_{RS}\}$.

Высказанное утверждение особенно важно для проблем отказоустойчивости систем управления. Оно означает приоритетность восстановления устойчивости и качества собственных движений M_S перед свойствами инвариантности M_{YSF} – характеристик каналов воспроизведения задания и подавления возмущений. Соответственно приоритетны свойства инвариантности M_{YSF} к любым возмущениям по отношению к селективной инвариантности M_{RS} – независимости (слабой зависимости) установившихся составляющих вынужденных движений к определенным возмущениям.

В соответствии с принципом последовательного раскрытия неопределенностей вводится понятие моделей различных рангов неопределенности $M(R)$ [2], [7].

Базовое множество подсистем представляет собой модель системы нулевого ранга неопределенности $M_S(0)$. Дополнение информации в виде несимметричного отношения на множестве дает топологию – модель первого ранга неопределенности $M_S(1)$. Дальнейшее раскрытие не-

определенности до модели второго ранга $M_S(2)$ заключается в выборе класса операторов и задании структур операторов подсистем. Наконец, в результате конкретизации значений параметров получаются полностью определенные модели третьего ранга $M_S(3)$.

Модель второго ранга $M(2)$ есть множество моделей третьего ранга $\{M(3)\}$, элементы которого различаются значениями параметров. На рис. 2, а условно изображен сигнальный граф G , причем подграф G' определен полностью. Если, например, выделенной дуге (c, d) соответствует передаточная функция $k_i/(T_i s + 1)$, параметры которой k_i, T_i принадлежат заданному множеству Q , то существует множество моделей с различными параметрами – это случай структурированной неопределенности.

Модель первого ранга $M(1)$ является множеством моделей второго ранга $\{M(2)\}$, у которых одна и та же топология, однако структуры операторов могут различаться. В общем случае элементы содержат неструктурированную неопределенность, например в виде аддитивных вариаций передаточных функций или частотных характеристик $\delta W_i(j\omega)$ дуг графа (рис. 2, б).

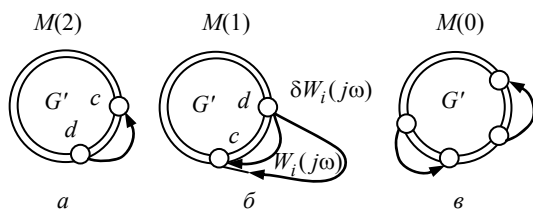


Рис. 2

Модель нулевого ранга $M(0)$ представляет собой множество моделей первого ранга $\{M(1)\}$ с различной топологией. В простейшем случае графы различаются местом включения одной дуги (рис. 2, в), например дуги, моделирующей регулятор или корректирующее устройство.

Понятие о рангах неопределенности моделей позволяет сформулировать принципиально важное для анализа и синтеза отказоустойчивых систем

Утверждение 2. Свойства системы сохраняют силу при повышении ранга модели.

Доказательство следует из того, что менее определенные модели низших рангов R' представляют собой множества более определенных моделей высших рангов R'' , что можно записать так:

$$M(R') \subseteq \{M(R'')\}; R' < R''.$$

Следовательно, свойства модели низшего ранга $M(R')$ присущи всему множеству моделей $M(R'')$.

Из утв. 2 следует ряд практически важных выводов:

1) проектирование СОУ реализует принцип эволюционного развития [5], [7], в соответствии с которым вначале формируется топология (причинно-следственная структура системы), после чего выбираются структуры операторов/алгоритмов подсистем и, наконец, оптимизируются параметры:

$$M(0) \rightarrow M(1) \rightarrow M(2) \rightarrow M(3);$$

2) при проектировании систем управления, их систем диагностирования и реконфигурирования первостепенное значение приобретает проблема формирования структур. Ошибки, допущенные при выборе структуры системы (топологии), не могут быть исправлены на последующих этапах выбора подсистем и, тем более, конкретизации значений их параметров;

3) результаты анализа и синтеза по моделям топологического ранга сохраняют силу при усложнении класса моделей. Например, результаты, полученные по моделям первого ранга $M(1)$, справедливы как для линейных, так и нелинейных, нестационарных, стохастических моделей.

Утверждение 3. Свойства системы сохраняют силу при усложнении класса модели.

Наименее сложным классом моделей являются линейные стационарные детерминированные конечномерные модели (класса ЛТИ) [5]. Они оказываются частными случаями (подмножествами) нелинейных, нестационарных, стохастических, распределенных (бесконечномерных) систем. Следовательно, свойства моделей класса ЛТИ в определенных ситуациях присущи моделям более сложных классов – нелинейных, нестационарных и т. д.

Из утв. 3 следует вывод о том, что теоретические основы отказоустойчивого управления целесообразно строить на базе моделей класса ЛТИ. Вывод дополнительно обоснован соображениями: 1) отказоустойчивость систем управления в режимах, адекватно описываемых линейными моделями, является необходимым (хотя и недостаточным) условием; 2) развитие неисправности может начинаться с малых вариаций, адекватно описываемых в рамках линейных моделей; 3) математика предлагает весьма мощный аппарат анализа линейных дифференциальных и разностных уравнений с постоянными коэффициентами;

4) в рамках теории управления разработаны специализированные методы и инструментальные средства анализа и синтеза систем этого класса;
5) выбор класса ЛТИ как основного класса, объяс-

няется и тем, что в рамках линейной теории возможно раздельное рассмотрение и формирование собственных и вынужденных составляющих движений.

СПИСОК ЛИТЕРАТУРЫ

1. Вавилов А. А. Структурный и параметрический синтез сложных систем: Учеб. пособие / ЛЭТИ. Л., 1979.
2. Вавилов А. А., Имаев Д. Х. Эволюционный синтез систем управления: Учеб. пособие / ЛЭТИ. Л., 1983.
3. Имитационное моделирование производственных систем / А. А. Вавилов, Д. Х. Имаев, В. И. Плескунин и др.; Под ред. А. А. Вавилова. М.: Машиностроение; Берлин: Veb Verlag Technik, 1983.
4. Шестопалов М. Ю. Системный подход к проектированию отказоустойчивых систем управления // Сб. докл. XVI Междунар. конф. по мягким вычислениям и измерениям "SCM-2013". СПб.; 23–25 мая 2013 г. СПб.: Изд-во СПбГЭТУ «ЛЭТИ». 2013. Т. 1. С. 14–19.
5. Теория автоматического управления: Учеб. для вузов / С. Е. Душин, Н. С. Зотов, Д. Х. Имаев и др.: Под ред. В. Б. Яковлева. 3-е изд., стер. М.: Высш. шк., 2009. 567 с.
6. Бусленко Н. П. Моделирование сложных систем. М.: Наука, 1968.
7. Шестопалов М. Ю. Методика построения систем реконфигурирования техническими объектами // Сб. докл. XVIII Междунар. заочн. науч.-практическая конф. «Технические науки – от теории к практике». Новосибирск: СибАК, 2013. С. 26–30.

M. Yu. Shestopalov

SYSTEMATIC APPROACH TO MATHEMATICAL MODELS CONSTRUCTION OF FAULT TOLERANT CONTROL OBJECTS

In the article as a fault tolerant control objects are observed technological processes control systems which are represented as a set of sub-systems with their functions and purposes for obtaining and processing the information, between which there is a transfer of information on connection-oriented channels.

Fault tolerant control objects, technological processes control

УДК 681.5

Ю. А. Королёв, А. А. Суворов,
В. Е. Герцман, И. В. Зименков

Проблемы реализации дистанционного управления измерительными средствами

Описываются основные принципы и проблемы, связанные с реализацией удаленного управления измерительными средствами наземной системы наблюдения и контроля космических запусков. Представлена архитектура построения программного комплекса дистанционного управления, примеры описания заданий на сеанс измерений.

Дистанционное управление, измерительные средства, режимы управления, сеанс измерений

При проведении запусков космических аппаратов необходимо обеспечить измерения на протяженных трассах полета при расположении измерительных пунктов (ИП) на значительном удалении

как от космодрома, так и от мест с развитой инфраструктурой жизнеобеспечения. Современные средства наблюдения и телеметрического контроля, а также комплексы сбора и передачи измерительной

информации должны функционировать в режиме дистанционного управления, что позволит сократить затраты на эксплуатацию и обслуживание удаленных ИП. Состав измерительных средств (ИС), для которых в первую очередь необходима реализация дистанционного управления, следующий:

- антенные системы и радиотелеметрические приемно-регистрирующие станции;
- оптико-электронные станции наблюдения и измерений;
- аппаратура систем единого времени.

Задача дистанционного управления измерительным средством сводится к классической задаче разработки автоматизированного объекта управления. Автоматизированным объектом управления традиционно называют пару $\langle A, O \rangle$, состоящую из управляющего автомата и объекта управления (ОУ) [1]. Для реализации группового управления несколькими ИС одновременно необходимо разработать автоматизированную систему дистанционного управления, предназначенную для решения следующих задач:

- проведение мониторинга и диагностики технического состояния ИС без участия технического персонала непосредственно на ИП;
- реализация централизованной подготовки и автоматической рассылки целеуказаний и входных заданий;
- контроль в реальном масштабе времени режимов работы измерительных и обеспечивающих систем ИП;
- выполнение программного или директивного изменения режимов работы ИС при подготовке и проведении сеанса измерений.

В соответствии с задачами автоматизации управления ИС, к разрабатываемой автоматизированной системе управления выдвигаются следующие основные требования:

- 1) возможность управления ИС в ручном, автоматическом и автоматизированном режимах;
- 2) информационное взаимодействие с ИС осуществляется по цифровой сети передачи данных, которая предоставляет стандартные услуги связи, но не является предметом разработки системы дистанционного управления;
- 3) автоматизированная система управления должна обеспечивать возможность группового управления несколькими ИС одновременно, а также предоставлять средства формирования и автоматического выполнения сценариев удаленного управления;

4) возможность простого горизонтального масштабирования автоматизированной системы управления другими объектами управления, аналогичными ИС с точки зрения указанных выше основных задач.

Далее в тексте два термина ОУ и ИС будут употребляться как синонимы.

Неотъемлемой частью современных ИС является цифровая вычислительная машина (центральный вычислитель), решающая задачи регистрации и обработки исходных измерений и формирования сигналов управления на исполнительные механизмы ИС с обеспечением заданного качества управления (по точности, устойчивости и быстродействию). На центральных вычислителях ИС установлены консоли управления, которые предоставляют программный интерфейс для отправки набора команд и запроса текущего состояния ИС. В соответствии с предъявляемыми требованиями дистанционное управление предлагается организовать следующим образом. Для подачи по сети передачи данных управляющего воздействия на консоли управления необходимо разработать программный комплекс дистанционного управления (ПК ДУ) и набор компонентов сетевого взаимодействия (по одному для каждого ИС). Такие компоненты необходимо установить на центральные вычислители измерительных средств для сопряжения единого сетевого управляющего интерфейса ПК ДУ с программными интерфейсами консолей управления.

Информационное взаимодействие между ПК ДУ и ОУ. При проектировании механизма информационного взаимодействия между ПК ДУ и ИС, необходимо выбрать:

- транспортный протокол;
- язык описания абстрактного синтаксиса и кодогенератор к нему;
- состав сетевого оборудования.

Поскольку каждое измерительное средство имеет свой уникальный набор управляющих команд, для каждого ИС должен быть разработан собственный прикладной протокол информационного взаимодействия.

На транспортном уровне целесообразно использовать один из наиболее широко используемых протоколов в стеке TCP/IP: TCP или UDP. В связи с тем, что ПК ДУ устанавливается на аппаратных средствах пункта дистанционного управления, соединенного с группой центральных вычислителей ИС, неизбежно пересечение потоков

данных от различных ИС. Вероятность потери UDP-пакетов при такой топологии будет достаточно велика, поэтому в данном случае предпочтительнее использовать в качестве транспортного протокола – протокол TCP, обеспечивающий надежную доставку данных.

Для эффективного кодирования и декодирования сообщений протокола информационного взаимодействия целесообразно использовать язык описания абстрактного синтаксиса данных и кодогенератор к нему [2]. Широко распространены языками описания данных, имеющими кодогенераторы с открытым исходным кодом для различных языков программирования, являются: ASN.1, Google Protocol Buffers, Apache Thrift. Для сравнения возможностей указанных языков использовались следующие критерии: возможности языка, качество генерируемого кода, компактность бинарного представления, количество поддерживаемых языков и качество документации. По результатам сравнения лучшие характеристики по совокупности критериев показал язык описания данных Google Protocol Buffers и его официальный кодогенератор protoc. Универсальность системы автоматизированного управления ИС к

типам доступных цифровых сетей связи (спутниковые, радиорелейные, волоконно-оптические) обеспечивается за счет использования стандартного оконечного оборудования для передачи данных – многопортовых коммутаторов и маршрутизаторов с портами Ethernet.

Режимы управления. Для осуществления дистанционного управления измерительным средством необходимо реализовать 3 режима: ручной, автоматизированный и автоматический. В режиме *ручного* управления доступ к центральным вычислителям измерительных средств осуществляется посредством стандартной системы удаленного доступа к компьютеру. Удаленное администрирование центрального вычислителя ИС (а также сетевых устройств, поддерживающих процесс удаленного управления) необходимо в следующих случаях:

- 1) для проверки функционирования центрального вычислителя и установленного на него программного обеспечения (удаленное администрирование);
- 2) для работы удаленного оператора непосредственно с консолью управления измерительным средством в случае возникновения нештатных ситуаций.

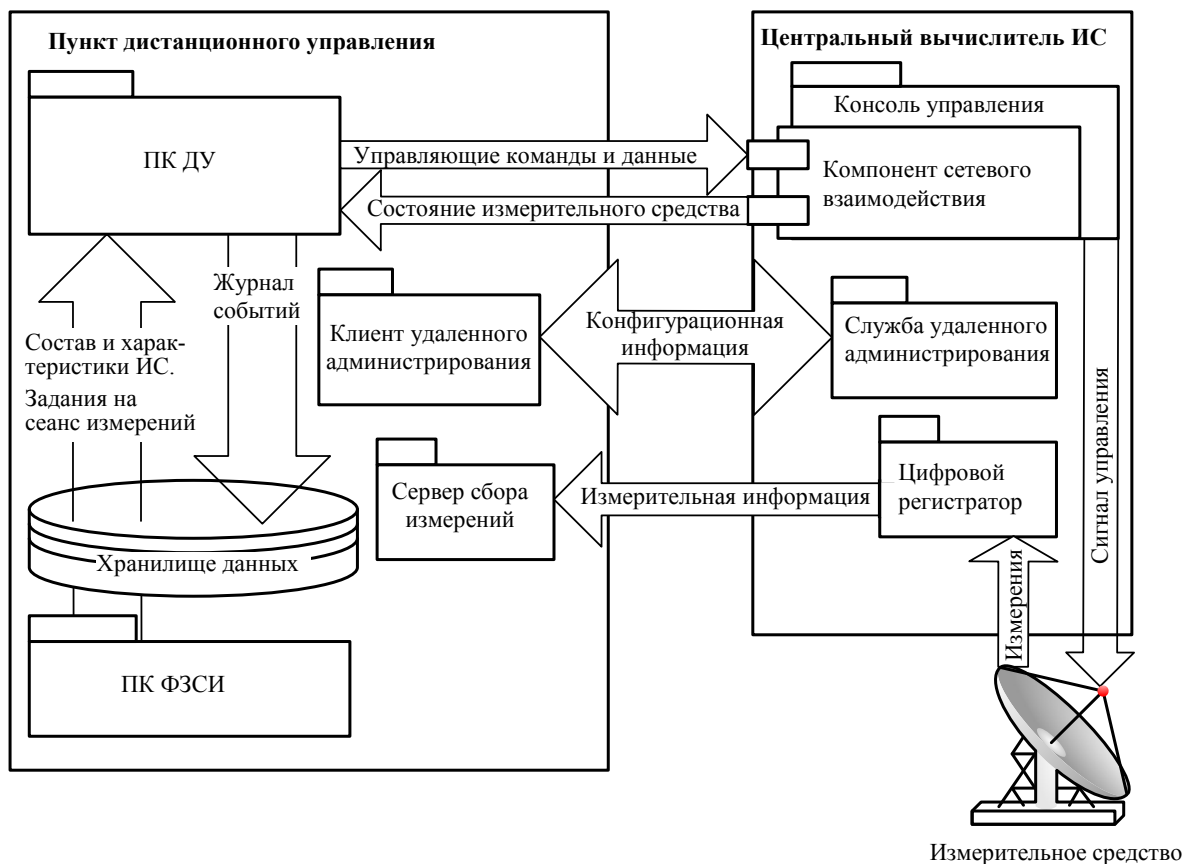


Рис. 1

Удаленное администрирование может быть реализовано с помощью сетевых протоколов удаленного доступа (RDP, VNC, SSH, SNMP) [3]. При этом для повышения скорости взаимодействия возможно применение специализированных аппаратных средств обеспечения удаленного доступа (IP-KVM передатчиков и IP-KVM приемников). В *автоматизированном* режиме оператор должен иметь возможность директивного изменения режимов работы ИС посредством отправки команд управления (уникальных для каждого ИС). В *автоматическом* режиме управление происходит за счет выполнения заранее подготовленного сценария, и участие оператора заключается лишь в запуске данного режима, мониторинге хода выполнения сценария и выполнения каких-либо действий в случае возникновения нештатных ситуаций.

Подход к дистанционному управлению ИС.

Для того чтобы управлять несколькими ИС одновременно в автоматическом режиме, необходим подход, при котором оператор может заранее записать алгоритм управления ИС. Формальное описание такого алгоритма управления измерительным средством будем называть *заданием на сеанс измерений*. ПК ДУ должен автоматически выполнять содержащийся в задании алгоритм одновременно для группы ИС.

Возлагаемые на автоматизированную систему управления задачи могут быть разделены на 2 группы, относящиеся к этапу формирования задания на сеанс измерений и к этапу дистанционного управления. В соответствии с таким разделением, в состав разрабатываемой системы должны входить два программных комплекса: программный комплекс дистанционного управления (ПК ДУ) и программный комплекс формирования заданий на сеанс измерений (ПК ФЗСИ), взаимодействующих друг с другом через разделяемое хранилище данных (рис. 1).

Для дистанционного управления каждым ИС необходимо разработать собственный протокол информационного взаимодействия, который может содержать несколько десятков команд. В самом простом случае алгоритм управления можно записать в терминах таких команд. Однако такое решение имеет ряд очевидных недостатков:

1) для того чтобы реализовать даже простейшую операцию, оператору необходимо будет записать достаточно сложный алгоритм. Например, даже простой поворот антенны в заданные коор-

динаты должен учесть следующие моменты: отправку команды на поворот, обработку квитанции об успешном принятии команды, обработку квитанции о невозможности выполнения операции, циклический опрос состояния антенны, учет таймаутов. Конечному пользователю системы такие сложности не нужны, ему необходимо только одно – послать команду на поворот и получить результат (успешно или нет);

2) даже при незначительном изменении протокола или логики работы ИС необходимо корректировать задания. В реальности – такие изменения неизбежны на протяжении всего срока эксплуатации автоматизированной системы управления.

В связи с вышеперечисленными недостатками алгоритм управления предлагается формировать в терминах высокоуровневых операций, которые будут выступать в качестве входных данных для управляющих автоматов. Каждая операция реализуется последовательностью команд, выполняемых автоматом на основе содержащегося в нем алгоритма. Жизненный цикл задания на сеанс измерений представлен на рис. 2.

Такое решение нивелирует минусы варианта с использованием команд из протокола информационного взаимодействия и имеет ряд дополнительных преимуществ:

1) пользователь формирует задание только из высокоуровневых операций, которых немного. В результате получается компактное задание, которое легко читать и понимать;

2) набор высокоуровневых операций гораздо стабильнее, чем их реализация с помощью протокола информационного взаимодействия с ИС. Поэтому при модернизации ИС, сохраненные задания, скорее всего, не придется изменять;

3) управляющие автоматы могут поддерживать единый входной формат представления задания, а не композицию структур из протокола информационного взаимодействия для каждого ИС сопровождаемую служебной информацией;

4) при использовании такого единого формата становится возможным унификация форм ввода данных.

Задание на сеанс измерений. Для реализации автоматизированного формирования и разбора задания на сеанс измерений его следует записывать с использованием специального языка разметки. Одним из наиболее широко используемых на сегодняшний день языков логической разметки является язык XML. Его распространение обу-

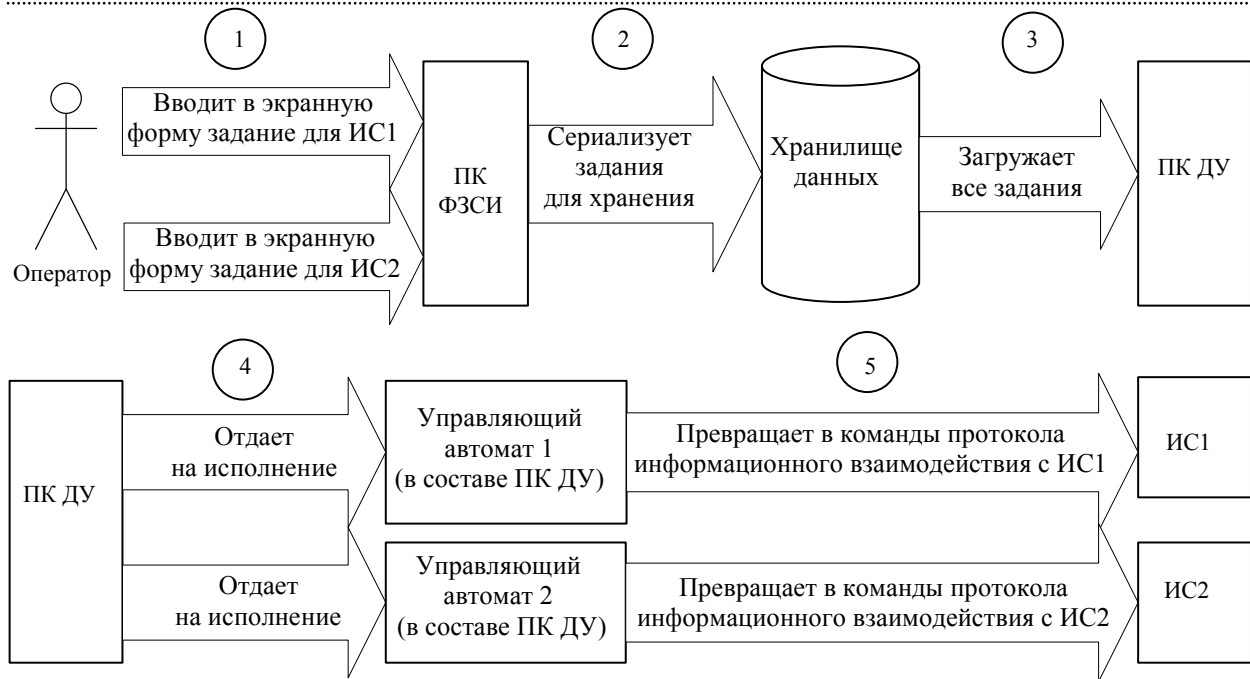


Рис. 2

словлено рядом причин, главными из которых являются: простой формальный синтаксис, удобство создания и обработки документов человеком, наличие большого количества программных библиотек для разбора и создания XML-документов. Таким образом, в качестве языка разметки для представления задания на сеанс измерений был выбран язык XML.

При создании программного обеспечения, осуществляющего обработку документов XML, традиционно используется XML Schema – язык описания структуры XML-документа, определяющий правила, которым должен подчиняться документ.

XML-схему для описания задания на сеанс измерений удобно разделить на 2 файла: файл со словарем операций и их параметров, файл с описанием структуры задания. Такие файлы создаются в отдельности для каждого ИС. Фрагмент словаря, содержащий определение параметра «Азимут» и операции «Съемка опорного ориентира», имеет следующий вид:

```

<xsd:simpleType name="azimuth">
  <xsd:annotation nic:unit="Угловая сек."
    nic:friendlyName="Азимут"/>
  <xsd:restriction base="xsd:integer">
    <xsd:minInclusive
      value="0"/>
    <xsd:maxInclusive
      value="1296000"/>
  </xsd:restriction>
</xsd:simpleType>

```

```

</xsd:simpleType>
<xsd:complexType
  name="shootingMilestones">
  <xsd:annotation nic:friendlyName="Съемка
    опорного ориентира"/>
    <xsd:attribute type="azimuth"
      name="startingAzimuth"/>
    <xsd:attribute
      type="frequency" name="frequency"/>
  </xsd:complexType>

```

Параметр операции представляет собой простой тип, возможно имеющий ограничения на область допустимых значений. Поскольку на основе словаря команд в ПК ФЗСИ будет осуществляться автоматическая генерация форм ввода параметров операций, каждый параметр необходимо снабжать аннотацией, содержащей его имя на русском языке и единицы измерения. Для описания структуры операции используется составной тип, который содержит в качестве атрибутов произвольное число определенных ранее параметров. Указанные выше определения параметров являются универсальными для систем управления антенными комплексами телеметрических измерений и теодолитами различных ОЭС.

Структура задания на сеанс измерений выглядит следующим образом:

```

<xsd:complexType
  name="abstractOperation">
  <xsd:attribute type="xsd:integer"
    name="id"/>

```



```

<xsd:attribute          type="xsd:integer"          type="shootingMilestones"/>
name="successId"/>                                     </xsd:choice>
<xsd:attribute          type="xsd:integer"          </xsd:extension>
name="unsuccessId"/>                                     </xsd:complexContent> </xsd:complexType>
<xsd:attribute          type="xsd:integer"          <xsd:element          name="task">
name="timeoutId"/>                                     <xsd:complexType> <xsd:sequence>
</xsd:complexType>                                     <xsd:element          max-
<xsd:complexType       name="operation">              Occurs="unbounded"    name="operation"
<xsd:complexContent>                                  type="operation"/>
  <xsd:extension base="abstractOperation">            </xsd:sequence>      </xsd:complexType>
    <xsd:choice minOccurs="1" max-
Occurs="1">
      <xsd:element
name="setWeatherData"
type="setWeatherData"/>
      <xsd:element
name="shootingMilestones"

```

Вначале определяется абстрактный тип «операция», содержащий набор атрибутов, с помощью которых можно формировать алгоритм управления с циклами и ветвлениями, а также корректно обрабатывать ошибки сетевого взаимодействия. К таким атрибутам относятся: идентификатор операции, номера следующей операции, если теку-

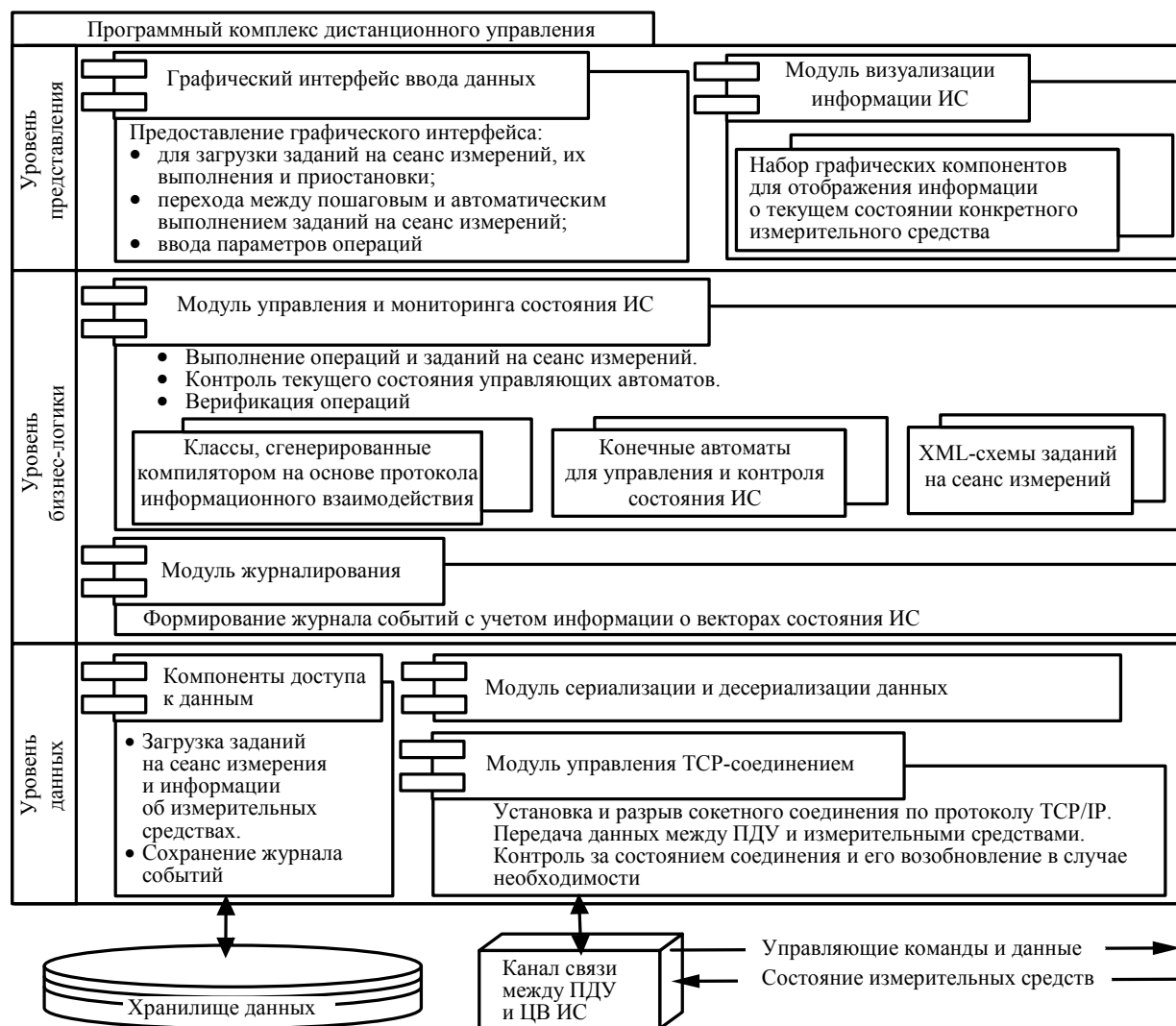


Рис. 3

щая выполнена: успешно / неуспешно / превышен таймаут ожидания. Тип `abstractOperation` наследуется типом `operation`, который представляет собой обертку над всеми возможными операциями конкретного ИС. Элемент `task` включает в себя неограниченную последовательность операций.

Архитектура ПК ДУ. Программный комплекс дистанционного управления имеет многоуровневую архитектуру (рис. 3). Все компоненты программного комплекса традиционно разделены на три логических уровня [4]: уровень данных, уровень бизнес-логики и уровень представления. Функциями *уровня данных* являются хранение и передача данных между различными поставщиками и потребителями информации.

Уровень бизнес-логики предоставляет объектную модель следующему слою и обеспечивает независимость уровня представления от модели хранения данных. *Уровень представления* осуществляет отображение данных и реализует интерфейс взаимодействия с пользователем. Ключевой модуль ПК – модуль управления и мониторинга состояния измерительных средств – получает на вход загруженные из хранилища данных задания на сеанс измерений. После верификации заданий на сеанс измерений на соответствие XML схемам, данный модуль начинает их выполнение, используя набор управляющих автоматов (по одному для каждого ИС). По мере выполнения заданий осуществляется контроль управляющих автоматов и уведомляется модуль отображения информации ИС об изменении значений векторов состояния измерительных средств.

Следует отметить, что программный комплекс дистанционного управления имеет расширяемую компонентную архитектуру. При необходимости дистанционного управления новым измерительным средством, соответствующие программные модули могут быть легко добавлены в ПК ДУ. Для этого программисту требуется реализовать два программных модуля, в которых будет заключена логика работы с конкретным ИС: управляющий автомат и визуализатор состояния ИС. Процесс добавления вышеописанных модулей не затрагивает логику работы других компонентов ПК ДУ, поэтому данное архитектурное решение предотвращает внесение ошибок в существующий программный код при наращивании дополнительной функциональности.

В общем случае технология применения автоматизированной системы управления ИС позволяет отказаться от постоянного использования квалифицированного обслуживающего персонала на удаленных периферийных ИП за исключением аварийных или других нештатных ситуаций. Все наукоёмкие задачи, требующие участия квалифицированных инженеров-испытателей, сосредоточены в вычислительном центре и относятся преимущественно к подготовительной стадии сеанса измерений. После корректно проведённой подготовки, автоматизированная система управления отработает с минимально возможным составом операторов, в большинстве штатных ситуаций – автоматически, без участия человека.

СПИСОК ЛИТЕРАТУРЫ

1. Поликарпова Н. И., Шалыто А. А. Автоматное программирование. 2-е изд. СПб.: Питер, 2011.
2. Fielding R. T. Architectural Styles and the Design of Network-based Software Architectures. PhD dissertation. University of California, Irvine, 2000.
3. Калужный А. Д., Табунщик Г. В., Онищенко В. Ф. Средства передачи информации в терминальных системах // Системы обработки информации. 2011. № 3. С. 39–47.
4. Архитектура информационных систем / Б. Я. Советов, А. И. Водяхо, В. А. Дубенецкий, В. В. Цехановский. М.: Издательский центр «Академия», 2012.

Yu. A. Korolev, A. A. Suvorov, V. E. Hertzman, I. V. Zimenkov

REMOTE CONTROL IMPLEMENTATION ISSUES FOR MEASUREMENT APPARATUS

The article describes the basic principles and problems associated with the implementation of the remote control for measurement apparatus. Software architecture of remote control system and samples of tasks for a measurement session are presented.

Remote control, measurement apparatus, control modes, measurement session



УДК 004.94, 004.85

Ю. В. Романенко

Технология разработки компьютерных обучающих средств для подготовки эксплуатирующего персонала космического ракетного комплекса

Описываются состав, функциональные модули, технология разработки и архитектура компьютерных обучающих средств для подготовки персонала космического ракетного комплекса.

Компьютерные обучающие системы, интерактивные обучающие комплексы, программные тренажеры, надежность оператора

Современные космические ракетные комплексы (КРК) представляют собой сложные технологические системы, управляемые боевыми расчетами (БР) посредством формирования команд и воздействий с пультавого оборудования или ручных операций на технике, часто представляющей объекты повышенной опасности. Существенная доля технологических нарушений при эксплуатации КРК не связана ни со старением оборудования, ни с его конструктивным несовершенством, а вызвана недостатками ведения технологического процесса, просчетами в системе обслуживания материально-технических ресурсов и так называемым «человеческим фактором». Сложность структуры космических систем и комплексов, опасность некоторых операций, неизбежная потеря персоналом практических навыков работы в периоды между реальными процессами, а также высокая стоимость ремонта оборудования определяют и высокую цену любых сбоев в работе или ошибок персонала систем КРК. Поэтому остро стоит вопрос создания адекватной системы обучения, способной быстро реагировать на изменяющиеся требования и условия эксплуатации сложных систем КРК. Особую важность приобретает ее качество, обеспечение постоянной готовности БР и эффективности их действий в нештатных ситуациях.

В последнее время в связи с бурным развитием компьютерных технологий появилась возможность создания и использования компьютерных обучающих средств (КОС). Применение КОС

дает следующие преимущества и возможности в подготовке личного состава БР по сравнению с традиционными видами подготовки:

- сохранение ресурса дорогостоящего и уникального оборудования с одновременным повышением уровня подготовленности эксплуатирующего персонала;
- эффективное комбинирование различных режимов обучения: теоретической и практической подготовки, самоподготовки, групповой отработки действий в составе расчетов и т. п.;
- обеспечение технологической основы для гибкого взаимодействия между обучаемыми и обучающими (инструкторами);
- немедленная реакция инструктора и корректирующие воздействия при неправильных действиях обучаемых;
- комбинирование различных форм представления информации (текстовой, графической, мультимедийной);
- динамическая адаптация режимов и курсов обучения к индивидуальным особенностям обучаемых;
- предоставление обучаемым права управлять объемом и очередностью порций учебного материала;
- ведение статистики и составление индивидуальных и групповых программ обучения с учетом результатов усвоения материала.

Комплексный подход к обучению персонала КРК, эксплуатирующего и обслуживающего сложные электронные системы, разработан в ОАО

«НИЦ СПб ЭТУ». Заключается он в применении на практике постоянно актуализируемого методического обеспечения и автоматизированных средств, реализующих принцип непрерывного обучения. Автоматизированные средства разрабатываются с использованием современных мультимедийных компьютерных технологий и выполняют следующие основные функции:

- моделирование ведения технологического процесса с отображением средств управления, соответствующих реальному операторскому интерфейсу и возможностью выполнения операций всех технологических режимов;
- моделирование аварийных и нештатных ситуаций технологического процесса;
- моделирование неисправностей оборудования и воздействия внешних факторов;
- протоколирование событий в системе и действий обучаемых.

Модульный принцип построения программ обучения позволяет конструировать индивидуальные учебные комплексы, наиболее полно отвечающие запросам потребителей в каждый момент времени. Процесс обучения организуется в соответствии со стандартом ГОСТ Р ИСО 10015–2007 «Менеджмент организации. Руководящие указания по обучению» (Москва, 2008). Указанный стандарт идентичен международному стандарту ISO 10015:1999 «Менеджмент качества. Руководящие указания по обучению». Указанные стандарты предполагают четыре стадии процесса обучения:

1. Определение потребностей (целей) обучения.
2. Проектирование (разработка) и планирование обучения.
3. Процесс обучения.
4. Оценка результатов обучения.

Целью создания КОС подготовки БР КРК является создание условий всестороннего изучения персоналом устройства, принципов функционирования и правил эксплуатации бортовых систем ракетносителей (РН), наземного технологического оборудования КРК, а также формирование, совершенствование и поддержание профессиональных навыков и умений персонала по выполнению технологических операций эксплуатации КРК. При этом КОС КРК должны обеспечивать:

- профессиональную подготовку персонала к выполнению действий, регламентированных эксплуатационной документацией, нормативно-техническими, распорядительными и уставными документами;
- поддержание профессиональных навыков обслуживающего персонала на уровне, обеспечивающем необходимое качество выполнения поставленных задач;

- контроль, анализ и оценку профессиональной подготовки, а при необходимости автоматическое документирование результатов выполнения функциональных обязанностей обслуживающего персонала;

- оценку профессионально-психологической пригодности обучаемых к работе по специальности.

Стадия проектирования (разработки) и планирования обучения предусматривает решение следующих задач:

- определение содержания программ;
- выбор существующих или разработка новых методов обучения;
- разработка методик, способов и средств обучения.

Выбор метода обучения зависит от конкретной цели: получить новые знания, сформировать умения, выработать установку на уровне мышления. Комплексный эффект достижения этих целей возможен только при сочетании нескольких методов обучения. Сегодня популярны активные методы, которые большое внимание уделяют именно практической основе передаваемых слушателям знаний, навыков и умений.

Процесс обучения и оценка результатов выполняются специалистами учебных центров непосредственно в воинских частях. В современных средствах обучения используются специальные макетные устройства и все возможности компьютерных программ. И хотя такие учебные комплексы требуют существенных финансовых вложений, они окупаются высоким качеством тренировки и приближенностью к реальным условиям работы. Однако количество обучаемых, допущенных одновременно к занятиям, ограничено и часто не удается комплексно отработать нештатные ситуации, а в некоторых случаях разработка и использование макетов реального оборудования невозможна по объективным причинам. Современные мультимедийные технологии и средства компьютерного моделирования позволяют создавать эффект реальности рабочих ситуаций и добиваться максимальной схожести виртуального оборудования и средств управления реальными системами.

Эффективность и результативность обучения персонала КРК в значительной степени определяются содержанием программно-методического комплекса, его соответствием эксплуатационным требованиям. Основными факторами, влияющими на содержание конкретного КОС, являются:

- этап обучения, для которого используется КОС;
- режим функционирования обучающего комплекса (форма занятия);

- приобретаемые знания и отрабатываемые навыки;
- требуемый уровень контроля действий обучаемого;
- уровень начальной подготовки обучаемых.

В общем случае в КОС КРК предусматриваются определенные этапы обучения.

1. *Изучение.* Цель этого этапа состоит в усвоении теоретических основ работы технических систем, приобретении глубоких знаний в области их устройства, порядка применения, особенностей функционирования и технического обслуживания.

2. *Индивидуальный тренаж.* Цель – приобретение, закрепление и совершенствование навыков управления техникой, выполнения технологических операций и т. д.

3. *Комплексный тренаж.* Цель – отработка слаженности действий подразделения (группы) при решении задач некоторого технологического процесса.

На каждом из перечисленных этапов обучения целесообразно применять различные виды КОС и обеспечиваемые ими формы занятий, в том числе:

- компьютерные учебные пособия;
- интерактивные обучающие комплексы;
- тренажеры.

Компьютерное учебное пособие (КУП) – это программно-методический комплекс, обеспечивающий возможность самостоятельно освоить учебный курс или его раздел. Оно объединяет свойства обычного учебника, справочника и вопросника. Компьютерное учебное пособие предоставляет обучаемому материалы по курсам и разделам курсов, выбранным руководителем занятий для индивидуальной подготовки, и вопросы для самопроверки и тестирования.

Интерактивные обучающие комплексы (ИОК) расширяют функции КУП возможностью оперировать с объектами определенного класса. Среда реализует отношения между объектами, операции над объектами и отношениями, соответствующие их определению, а также обеспечивает наглядное представление объектов и их свойств. Такие программы позволяют реализовать обучающие комплексы для операторов оборудования, требующего выполнения, например, механических операций.

Контент КУП и ИОК включает следующие элементы:

- содержание, глоссарий, интерактивный текстовый контент с ссылочной структурой;

- фотографии и двумерные статические изображения;
- динамические двумерные изображения, ролики;
- динамические трехмерные изображения, модели и анимационные фрагменты;
- видеоролики работы с реальным оборудованием;
- звуковое сопровождение;
- вопросы самопроверки;
- программы обучения, группы обучения, конфигурации учебных модулей;
- тестовые задания, статистику ответов, протоколы обучения и проверки знаний.

Типовая технология разработки КУП и ИОК состоит из нескольких этапов:

- информационное обследование, сбор и обработка исходных данных, фотографирование реального оборудования и помещений, оцифровка конструкторских и эксплуатационных документов;
- анализ исходных данных, разработка учебного контента, наполнение текстовых разделов КОС;
- обработка фотографий, разработка статических графических изображений;
- разработка сценариев динамических двумерных изображений и трехмерных роликов;
- разработка двумерных динамических изображений (flash-роликов);
- разработка трехмерных моделей оборудования, создание и генерация трехмерных роликов;
- разработка трехмерных роликов с операциями использования оборудования по назначению и его обслуживания;
- разработка звуковых сценариев, озвучивание контента;
- сборка интерактивно-обучающего комплекса, тестирование, отладка, проверка контента;
- разработка эксплуатационной документации КУП (ИОК);
- сборка и наладка дистрибутива;
- передача изделия заказчику, пуско-наладочные работы, обучение эксплуатирующего персонала, ввод в эксплуатацию.

Аппаратно-программные тренажеры предназначены для максимально достоверной имитации технологии работы с оборудованием и отработки практических навыков, в том числе при групповых операциях. Такие тренажеры представляют собой специальное программное обеспечение, работающее на одном или нескольких персональных компьютерах, объединенных общей сетью, и реализующее математическую модель изменения состояния системы в зависимости от действий

операторов и задаваемых внешних факторов. Обучаемый взаимодействует с виртуальными устройствами, внешний вид и технология работы с которыми повторяют реальные устройства. Тренажеры для индивидуального и группового обучения моделируют штатные технологические процессы и нештатные ситуации. Программный тренажер легко перенастраивается при модернизации оборудования, что позволяет реализовать практически любые по сложности эксперименты в соответствии с методиками отработки разных нештатных ситуаций.

Тренажеры разрабатываются на основе современных компьютерных и мультимедийных технологий и реализуют следующие основные функции:

1. *Имитация системы.* Реализуется имитация рабочего места оператора и его действий путем интерактивного взаимодействия с компьютером. Имитация рабочего места имеет две составляющие: визуальную и функциональную. Визуальные образы рабочих мест создаются с помощью графических программ. Для реализации движений, динамических процессов или физических явлений используется программная анимация. Функциональная имитация объектов выполняется на основе модели системы. Она представляет собой совокупность параметров, характеризующих состояние системы в каждый момент времени и математические законы их изменения. Законы отражают физические процессы, которые происходят в системе при работе с ней операторов и при изменении ее состояния.

2. *Взаимодействие обучаемого с объектами компьютерного тренажера.* Выполняется мышкой или через сенсорные экраны.

3. *Моделирование штатного технологического процесса и нештатных ситуаций.* Компьютерный тренажер обеспечивает возможность обучения и оценки операторов при выполнении ими операций, увязанных в единый технологический процесс. В соответствии с заложенной моделью тренажер реагирует на действия оператора и на заданные внешние факторы, изменяя состояния устройств управления. Помимо штатных технологических процессов в тренажере реализуется возможность моделирования условий возникновения нештатных ситуаций, отработки операторами правильной последовательности действий для выхода из них и приведения системы в штатное состояние.

4. *Контроль действий обучаемых, накопление и анализ статистики работы операторов.* Контроль заключается в постоянном автоматическом мониторинге тренажером действий обучаемого.

В случае неверных действий обучаемого возможны различные сценарии реакции тренажера: приостановка процесса обучения, подсказка с продолжением обучения после правильного действия оператора и др. Альтернативным методом контроля является визуальный контроль действий обучаемого преподавателем, который в режиме реального времени или после сеанса обучения анализирует результаты и определяет успешность действий обучаемого.

5. *Информационная поддержка обучаемого.* Эта функция заключается в выдаче оператору дополнительной информации в интерактивном и контекстном режимах.

6. *Автоматическое выполнение тренажером действий операторов.* При одновременном обучении нескольких операторов в рамках одного технологического процесса возможна ситуация, при которой состав обучаемых будет неполным. В этом случае компьютерный тренажер автоматически выполняет действия отсутствующих операторов.

7. *Групповое обучение* (одновременная работа нескольких номеров боевых расчетов). В сложных технологических процессах, где участвуют несколько номеров боевых расчетов, важна не только правильная работа каждого оператора, но и синхронность их действий. Компьютерный тренажер обеспечивает сетевой вариант работы, при котором расчеты работают за отдельными компьютерами, соединенными в общую сеть. Тренажер выполняет синхронизацию работы нескольких компьютеров в рамках одного сеанса, имитирующего процесс несения боевого дежурства и выполнения боевых задач.

Типовая технология разработки компьютерного тренажера включает следующие этапы:

- сбор и анализ исходных данных для разработки сценариев работы тренажера и графических ресурсов;
- формализация и описание рабочих мест операторов боевых расчетов;
- разработка сценария работы тренажера, описание действий, выполняемых операторами, и реакций системы в штатном технологическом процессе и нештатных ситуациях;
- разработка виртуальных рабочих мест операторов;
- реализация сценария работы тренажера, динамического поведения элементов управления и индикации в зависимости от действий операторов в штатном технологическом процессе и нештатных ситуациях;

- разработка двумерных и трехмерных динамических изображений, изготовление и монтаж видеороликов;

- разработка конструкторской и эксплуатационной документации;

- тестирование, отладка, корректировка исходного кода, сборка дистрибутива;

- испытания и корректировка программных модулей и документации по результатам испытаний, пуско-наладочные работы, передача заказчику, ввод в эксплуатацию, обучение персонала.

Значительную часть процесса разработки КОС занимает этап анализа предметной области и разработки наполнения КУП, ИОК и программных тренажеров. Определяющим фактором успешного выполнения этапа анализа и разработки КОС в целом является квалификация аналитиков – экспертов в предметной области. Именно специалисты предметной области выполняют анализ конструкторской и эксплуатационной документации, формальное описание технологических процессов, разработку сценариев демонстрационного и обучающего контента.

Современные КОС разрабатываются с использованием трехуровневой архитектуры, включающей клиентское и серверное приложения и базу данных. Клиентское приложение обеспечивает интерфейс пользователя и предназначено для отображения экранных форм, ввода и отображения данных, необходимых для работы КОС. Посредством клиентского приложения отображаются учебные материалы, преподаватель осуществляет настройку групп обучаемых и профилей обучения, проводятся тесты и проверки знаний. Серверное приложение предназначено для управления КОС, обработки данных, необходимых для работы клиентского приложения, обработки результатов (статистики) обучения и тестирования, управления правами доступа в соответствии с назначенными профилями обучения. Серверное приложение может также осуществлять выгрузку, передачу и загрузку данных на узлах КОС в случаях, когда они представляют собой территориально распределенную систему. Через серверное приложение при необходимости осуществляется интеграция различных типов КОС в единую систему обучения. База данных предназначена для хранения всей совокупности данных, необходимых для работы КОС. Она обеспечивает механизмы доступа к данным на чтение и запись, контроль целостности и непротиворечивости данных. База данных представляет собой совокуп-

ность данных, хранящихся в структурированном виде и в виде файлов. В структурированном виде хранятся справочники и оперативные данные, список персонала, составы учебных групп и профили (программы) обучения. В файловой системе хранятся учебные материалы, ресурсы (чертежи, видеофрагменты, текстовые страницы), сценарии работы тренажерных комплексов и т. д. Такая архитектура позволяет обеспечить одновременную работу нескольких пользователей, разделение специального программного обеспечения руководителя обучения и обучаемых при сохранении принципа централизованной обработки и хранения данных серверным приложением и базой данных. Подробно архитектурные аспекты разработки КОС рассмотрены в [1], [2].

Описанная технология разработки КОС обеспечивает возможность создания учебно-тренировочных средств для теоретической и практической подготовки специалистов в различных областях деятельности, обучать их как стандартным процедурам управления техникой, так и действиям в нештатных (аварийных) ситуациях. Представленная технология предполагает использование платформенного подхода, при котором КОС создаются на базе ранее разработанных и отлаженных технических решений. Использование платформы позволяет значительно сократить сроки и трудоемкость разработки конкретного КОС.

Компьютерные обучающие средства позволяют готовить специалистов к работе со сложными механизмами, оборудованием и приборами, изучать технологические процессы, в которых одновременно участвуют несколько операторов. Внедрение систем обучения и тренинга персонала позволяет снижать аварийность на производственных участках и в связи с этим уменьшать простои и разного рода затраты и компенсации. Полное использование персоналом знаний, навыков и умений, полученных в результате обучения, позволяет достаточно быстро окупать инвестиции, вложенные в КОС. Через обучение достигается и повышение способности персонала адаптироваться к изменяющимся условиям работы и все более жестким требованиям к эксплуатации сложных технических комплексов. Немаловажным аспектом обучения персонала и повышения его квалификации является обеспечение требуемого уровня надежности функционирования дорогостоящего оборудования за счет минимизации влияния «человеческого фактора».

СПИСОК ЛИТЕРАТУРЫ

1. Филатова Н. Н., Вавилова Н. И., Ахремчик О. Л. Мультимедиа-тренажерные комплексы для технического образования // Educational Technology & Society. 2003. № 6 (3). С. 164–186.
2. Романенко Ю. В. Архитектура компьютерного тренажера для обучения и аттестации операторов при управлении технологическими процессами и возникновении нештатных ситуаций // Изв. СПбГЭТУ «ЛЭТИ». 2007. Вып. 2. С. 70–80.

Yu. V. Romanenko

COMPUTER TRAINING TOOLS DEVELOPMENT TECHNOLOGY FOR LEARNING OF OPERATING PERSONNEL SPACE ROCKET COMPLEX

Structure, functional modules, development technology and architecture of computer-based simulators for operators of space-rocket complex are described in the article.

Computer training systems, interactive E-learning systems, computer-based simulator, operator

УДК 004.942, 519.876.5

М. Г. Пантелеев, А. В. Рюмин, С. В. Лебедев

Программный комплекс моделирования пусков для подготовки специалистов-анализаторов в области ракетной техники

Рассматриваются вопросы создания программного комплекса имитационного моделирования пусков ракет для подготовки специалистов-анализаторов результатов испытательных пусков. Приводятся математические модели полета ракет, используемые при разработке комплекса, методика моделирования нештатных и аварийных ситуаций. Представлена архитектура программного комплекса.

Моделирование пусков, математическая модель, программный комплекс, архитектура

Важнейшим этапом освоения новых образцов ракетно-космической техники (РКТ) является проведение пусковых испытаний с последующим анализом летно-технических характеристик (ЛТХ) изделий. Поддержание высокого уровня подготовки специалистов-анализаторов (СА) результатов испытательных пусков предполагает регулярные тренинги. В условиях достаточно редких реальных пусков и неприемлемости (по экономическим соображениям) натурного воспроизведения нештатных и аварийных ситуаций такую задачу целесообразно решать с использованием средств имитационного моделирования пусков.

Основные требования к программному комплексу моделирования пусков:

- возможность моделирования различных типов изделий в широком диапазоне начальных условий (состояние атмосферы, модель Земли и др.) с целью изучения их влияния на процесс полета;
- возможность моделирования нештатных и аварийных пусков за счет внесения различных (внешних и внутренних) возмущающих факторов;
- открытость и расширяемость.

Базовая математическая модель. В основу разработки программного комплекса положена математическая модель движения ракеты как тела

переменной массы и позволяющая определять параметры ее состояния в любой момент времени [1], [2]. Модель учитывает изменения массогабаритных и аэродинамических характеристик, параметров атмосферы и силы тяжести Земли.

Общее уравнение движения центра масс ракеты. Модель движения центра масс в траекторной системе координат (СК) описывается следующим уравнением [1], [3]:

$$\dot{\mathbf{V}} + \boldsymbol{\omega}_T \times \mathbf{V} = \frac{1}{m} (\mathbf{A}_{12} \cdot \mathbf{P} - \mathbf{G} - \mathbf{A}_{11}^T \cdot \mathbf{R} + \mathbf{A}_{12} \cdot \mathbf{P}_{\text{упр}}) - \mathbf{a}_{\text{пер}} - \mathbf{a}_{\text{кор}},$$

где $\boldsymbol{\omega}_T = [\omega_x \ \omega_y \ \omega_z]^T$ – вектор угловых скоростей траекторной системы координат относительно стартовой; $\mathbf{P}$ – вектор силы тяги ракетного двигателя; $\mathbf{G}$ – вектор характеризующий текущий вес ракеты; $\mathbf{R}$ – вектор аэродинамических сил; $\mathbf{P}_{\text{упр}}$ – вектор управляющих сил; $\mathbf{A}_{11}, \mathbf{A}_{12}$ – матрицы перехода; $\mathbf{a}_{\text{кор}}$ – кориолисово ускорение; $\mathbf{a}_{\text{пер}}$ – переносное ускорение.

Тяга двигателя в связанной системе координат определяется выражением [2]

$$\mathbf{P} = [1 \ 0 \ 0]^T \times [\dot{m} W_{\text{отн}} + S_{\text{мид.соп}} (p_d - p_a)],$$

где $\dot{m}$ – секундный расход топлива; $W_{\text{отн}}$ – относительная скорость истечения газа; $S_{\text{мид.соп}}$ – критическое сечение сопла; p_a – атмосферное давление; p_d – давление газа на срезе сопла.

Текущий вес ракеты в траекторной СК описывается следующей формулой:

$$\mathbf{G} = [0 \ -m(t)g \ 0]^T.$$

Ускорение свободного падения с учетом несферичности Земли до второй гармоники полинома Лежандра имеет вид

$$g = \frac{fM}{r^2} \left[1 + \frac{3}{2} \frac{\mu}{M} \frac{R_3^2}{r^2} (1 - 3 \sin^2 \varphi_{\text{ГЦ}}) - q \frac{r^3}{R_3^3} \cos^2 \varphi_{\text{ГЦ}} \right],$$

где $f = 6.67428 \cdot 10^{-11} \text{ м}^3 \cdot \text{с}^{-2} \cdot \text{кг}^{-1}$ – постоянная всемирного тяготения; $\mu = 0.0011 M$; $M = 5.9736 \cdot 10^{24} \text{ кг}$ – масса Земли; $q = 0.003468$ – безразмерный параметр фигуры Земли, равный отношению ускорения центробежной силы к ускорению свободного падения в плоскости экватора; $R_3 = 6\,378\,245 \text{ м}$ – радиус экватора Земли.

Текущая масса ракеты равна:

$$m(t) = m_0 - \int_0^t \dot{m} dt,$$

где m_0 – начальная масса; $\dot{m}$ – секундный расход топлива.

Аэродинамическая сила, действующая на ракету, равна [2], [3]:

$$\mathbf{R} = [R_x \ R_y \ R_z]^T = q S_{\text{мид}} [c_x \ c_y \ c_z]^T,$$

где c_x, c_y, c_z – аэродинамические коэффициенты, определяемые в общем случае следующими зависимостями:

$$c_x = f(M, h); \ c_y = f(\alpha, M); \ c_z = f(\beta, M).$$

На практике аэродинамические коэффициенты задаются численно (как правило, таблично) либо рассчитываются отдельным подмодулем.

Управляющая сила записывается следующим образом [3]:

$$\mathbf{P}_{\text{упр}} = [0 \ \sin \delta_{\text{I-II}} I \ \sin \delta_{\text{I-IV}}]^T P,$$

где $\delta_{\text{I-III}}, \delta_{\text{II-IV}}$ – углы отклонения управляющего органа в соответствующих плоскостях.

Для идеальной системы управления можно принять:

$$\delta_{\text{I-III}} = \varphi_{\text{пр}} - \varphi,$$

$$\delta_{\text{II-IV}} = \psi_{\text{пр}} - \psi.$$

Программное управление моделируется заданием углов тангажа и рысканья как функций времени:

$$\varphi_{\text{пр}}, \psi_{\text{пр}} = f(t).$$

Кориолисово ускорение определяется выражением

$$\mathbf{a}_{\text{кор}} = 2(\boldsymbol{\Omega} \times \mathbf{V}),$$

где $\boldsymbol{\Omega} = \Omega_3 \cdot [0 \ 1 \ 0]$ – вектор угловой скорости вращения Земли; Ω_3 – угловая скорость вращения Земли.

Переносное ускорение

$$\mathbf{a}_{\text{пер}} = \boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times \mathbf{r}).$$

Матрица $\mathbf{A}_{11}$ перехода из траекторной СК к скоростной имеет вид

$$\mathbf{A}_{11} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma_a & -\sin \gamma_a \\ 0 & \sin \gamma_a & \cos \gamma_a \end{bmatrix}.$$

При полете без крена (т. е. при $\gamma_a = 0$) имеем:

$$\mathbf{A}_{11} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Матрица $\mathbf{A}_{12}$ перехода из связанной СК в траекторную имеет вид

$$\mathbf{A}_{12} = \begin{bmatrix} \cos \alpha \cos \beta & -\sin \alpha \cos \beta & \sin \beta \\ \sin \alpha \cos \gamma_a + \cos \alpha \sin \beta \sin \gamma_a & \cos \alpha \cos \gamma_a - \sin \alpha \sin \beta \sin \gamma_a & -\cos \beta \sin \gamma_a \\ \sin \alpha \sin \gamma_a - \cos \alpha \sin \beta \cos \gamma_a & \cos \alpha \sin \gamma_a + \sin \alpha \sin \beta \cos \gamma_a & \cos \beta \cos \gamma_a \end{bmatrix}.$$

При полете без крена имеем:

$$\mathbf{A}_{12} = \begin{bmatrix} \cos \alpha \cos \beta & -\cos \beta \sin \alpha & \sin \beta \\ \sin \alpha & \cos \alpha & 0 \\ -\cos \alpha \sin \beta & \sin \alpha \sin \beta & \cos \beta \end{bmatrix}.$$

Уравнения вращательного движения вокруг центра масс. При моделировании вращательного движения учитываются: момент, создаваемый эксцентриситетом тяги двигателя, момент аэродинамических сил и момент управляющих сил.

Геометрические параметры каждой (i -й) субракеты включают: длину l_i , диаметр d_i , положение центра давления c_{di} и др.

В общем виде уравнение вращательного движения имеет вид [1]–[3]

$$\dot{\boldsymbol{\omega}} = \mathbf{A}_I^{-1} \sum \mathbf{M} - \mathbf{A}_I^{-1} \mathbf{A}_\omega \mathbf{A}_I \boldsymbol{\omega},$$

где $\dot{\boldsymbol{\omega}} = [\dot{\omega}_x \quad \dot{\omega}_y \quad \dot{\omega}_z]^T$ – вектор производных

угловых скоростей; $\boldsymbol{\omega} = [\omega_x \quad \omega_y \quad \omega_z]^T$ – вектор

угловых скоростей связанной СК относительно стартовой; $\sum \mathbf{M} = [\sum M_x \quad \sum M_y \quad \sum M_z]^T$ –

вектор суммарных моментов всех внешних сил.

Тензор инерции $\mathbf{A}_I$ в общем случае имеет вид

$$\mathbf{A}_I = \begin{bmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{yx} & I_y & -I_{yz} \\ -I_{zx} & -I_{zy} & I_z \end{bmatrix}.$$

Считая ракету осесимметричным телом, получим:

$$\mathbf{A}_I = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix}.$$

Матрица $\mathbf{A}_\omega$ равна:

$$\mathbf{A}_\omega = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}.$$

Зная величины моментов внешних сил и моментов инерции, можно найти угловые скорости $\omega_x, \omega_y, \omega_z$.

Суммарный действующий на ракету момент определяется суммой моментов, создаваемых аэродинамическими и управляющими силами:

$$\sum \mathbf{M} = \mathbf{M}_{\text{аэр}} + \mathbf{M}_{\text{упр}}.$$

Моменты аэродинамических сил можно записать так:

$$\mathbf{M}_{\text{аэр}} = \begin{bmatrix} M_{x\text{аэр}} \\ M_{y\text{аэр}} \\ M_{z\text{аэр}} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & c_d & 0 \\ 0 & 0 & c_d \end{bmatrix} \cdot \begin{bmatrix} R_x \\ R_y \\ R_z \end{bmatrix}.$$

Точкой приложения управляющих сил считается хвостовая часть текущей (i -й) субракеты. Таким образом, плечо этих сил равно $(l_i - c_{mi})$, где c_{mi} – положение центра масс i -й субракеты. Для расчета управляющего момента используется проекция управляющих сил на ось OY (перпендикулярную продольной оси ракеты). Тогда момент управляющих сил принимает вид

$$\mathbf{M}_{\text{упр}} = \begin{bmatrix} M_{x\text{упр}} \\ M_{y\text{упр}} \\ M_{z\text{упр}} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & l - c_m & 0 \\ 0 & 0 & l - c_m \end{bmatrix} \cdot \begin{bmatrix} P_{x\text{упр}} \\ P_{y\text{упр}} \\ P_{z\text{упр}} \end{bmatrix}.$$

Тогда

$$\dot{\boldsymbol{\omega}} = \mathbf{A}_I^{-1} (\mathbf{M}_{\text{аэр}} + \mathbf{M}_{\text{упр}}) - \mathbf{A}_I^{-1} \mathbf{A}_\omega \mathbf{A}_I \boldsymbol{\omega}.$$

Связь производных $\dot{\phi}, \dot{\psi}, \dot{\gamma}$ и угловых скоростей $\omega_x, \omega_y, \omega_z$ можно описать с использованием формул перехода из нормальной земной СК в связанную СК:

$$\dot{\phi} = \omega_y \sin \gamma + \omega_z \cos \gamma;$$

$$\dot{\psi} = \frac{1}{\cos \phi} (\omega_y \cos \gamma + \omega_z \sin \gamma);$$

$$\dot{\gamma} = \omega_x - \operatorname{tg} \phi (\omega_y \cos \gamma + \omega_z \sin \gamma).$$

Углы атаки и скольжения могут быть найдены по формулам

$$\alpha = \varphi - \theta;$$

$$\beta = \Psi - \psi.$$

Связь между прямоугольными и сферическими геоцентрическими координатами выражается соотношениями [4]

$$x_3 = r \cos \varphi_{ГЦ} \sin \lambda;$$

$$y_3 = r \sin \varphi_{ГЦ};$$

$$z_3 = r \cos \varphi_{ГЦ} \cos \lambda.$$

где $\varphi_{ГЦ}$ и λ – геоцентрическая широта и долгота ракеты соответственно.

Внесение возмущений. Для изучения влияния различных факторов на процесс полета и моделирования нештатных и аварийных ситуаций в модель можно вносить случайные возмущения. В общем случае внесение возмущений описывается следующей зависимостью [1], [2]:

$$\tilde{f}(x, y) = f(x, y) + \xi(t),$$

где $\xi(t)$ – случайная составляющая, формируемая генератором случайных чисел в соответствии с заданным законом распределения (например, нормальным).

- проведение заданного числа сеансов моделирования для выбранного пуска;
- статистическую обработку и отображение результатов моделирования.

Начальные данные модельного пуска включают: аэромассогабаритные характеристики (АМГХ) изделия, модели атмосферы и Земли, описание точек старта и цели, циклограмму пуска, тангажную программу и др. Кроме того, для ряда параметров модели могут задаваться случайные возмущения, описанные выше. Для задания возмущения выбирается закон распределения случайной величины и задаются ее параметры. В частности, случайные составляющие могут вноситься в секундный расход топлива, главные моменты инерции, параметры атмосферы, времена выдачи и отработки команд циклограммы и др.

Поскольку реальные значения случайных составляющих в различных сеансах моделирования варьируются в пределах заданного распределения, на этапе моделирования это может приводить к некорректным ситуациям. Например, вследствие случайной задержки отработки команды включения двигательной установки (ДУ) очередной ступени команда на изменение угла тангажа может быть выдана при выключенной ДУ. Для изделий, использующих газодинамиче-

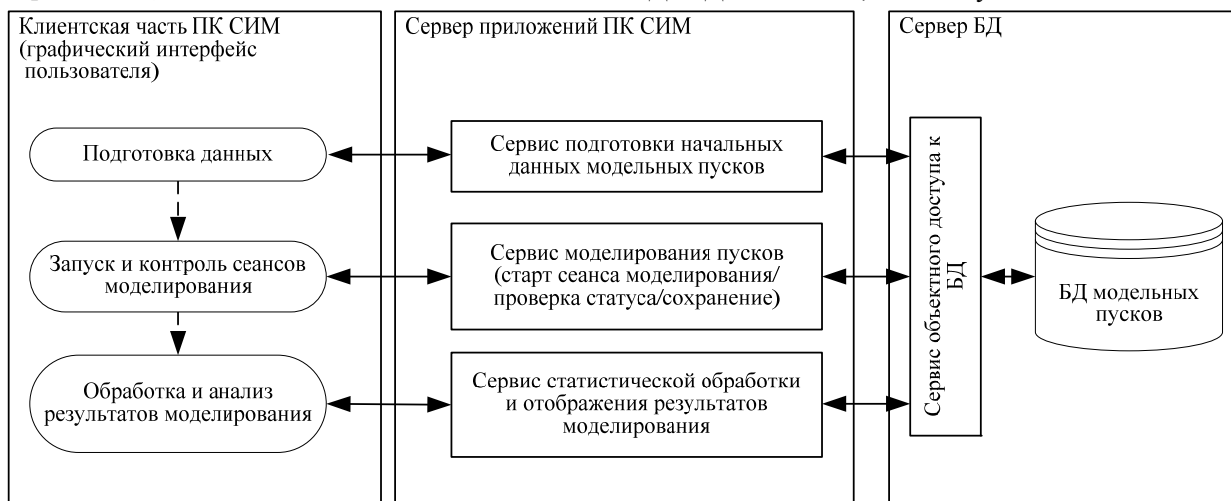


Рис. 1

Архитектура и функционирование программного комплекса моделирования. Программный комплекс ситуационно-имитационного моделирования пусков (ПК СИМ) построен в трехзвенной архитектуре (рис. 1) и поддерживает следующие основные функции:

- подготовку начальных данных моделируемого пуска;

ские рули, такая ситуация является недопустимой. Для исключения подобных ситуаций на этапе подготовки начальных данных реализован их семантический контроль с учетом предельных случайных отклонения значений параметров.

Моделирование выполняется по тактам, длительность такта задается пользователем. Для вычисления в каждом такте вектора состояния объекта на основе рассмотренной выше математической

модели используется численное интегрирование методом Рунге–Кутты 4-го порядка. Вектор состояния наряду с траекторными параметрами (координаты, скорость) в различных системах координат включает также углы тангажа, рысканья, вращения, наклона траектории, атаки и скольжения.

Кроме того, в процессе моделирования должны обрабатываться различные события, например, команды циклограммы и тангажной программы. Обработка таких событий реализуется специальными алгоритмами, учитывающими специфику конкретных событий, в частности, исполнительных устройств, отрабатывающих соответствующую команду.

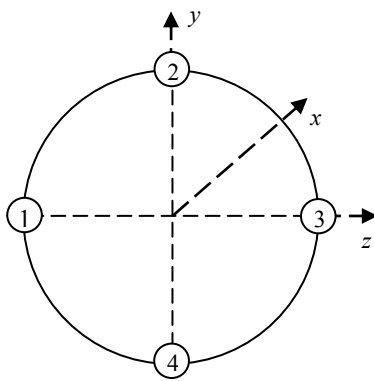


Рис. 2

Рассмотрим в качестве примера моделирование нештатной ситуации, вызванной одновременностью срабатывания пирозамков при отработке команды отделения ступени. Расположение точек их крепления на корпусе ракеты представлено на рис. 2.

Состав возникающих возмущающих моментов определяется последовательностью срабатывания пирозамков, соответствующих различным точкам крепления, следующим образом:

$(t_1 - t_3) = \Delta_y \rightarrow M_y$ – момент вращения по рысканью;

$(t_2 - t_4) = \Delta_z \rightarrow M_z$ – момент вращения по тангажу;

$$\left. \begin{aligned} (t_1 - t_2) &= \Delta_x \rightarrow M_x \\ (t_2 - t_3) &= \Delta_x \rightarrow M_x \\ (t_3 - t_4) &= \Delta_x \rightarrow M_x \\ (t_4 - t_1) &= \Delta_x \rightarrow M_x \end{aligned} \right\} \begin{array}{l} \text{момент вращения} \\ \text{по крену;} \end{array}$$

Для создания нештатной ситуации требуемого типа необходимо в процессе моделирования обеспечить соответствующие возмущающие моменты

при отделении ступени. Для этого на этапе подготовки данных задается соответствующая последовательность срабатывания пиропатронов и случайные времена задержки этих событий относительно соответствующей команды циклограммы.

На этапе выполнения сеанса моделирования конкретное время срабатывания каждого пиропатрона определяется с учетом случайной задержки относительно выдачи команды циклограммы. Время действия возмущающих моментов определяется интервалами между этими событиями. Таким образом, в различных сеансах моделирования одного пуска при одинаковом порядке возникновения возмущающих моментов их времена действия будут отличаться.

Принцип работы сервиса моделирования пусков иллюстрируется рис. 3.

Цикл моделирования начинается с обновления модельного времени в соответствии с заданным тактом моделирования. В начале каждого цикла проверяется наличие команд циклограммы и тангажной программы в текущем такте. Проверка выполняется с учетом возможных случайных отклонений времени выдачи команд. Далее в значения параметров модели, для которых были заданы случайные составляющие, вносятся соответствующие возмущения и рассчитывается новое состояние объекта. Обновление конфигурации (т. е. переход к новой субракетке) выполняется в тактах, когда произошло полное отделение ступени. Вычисленное новое состояние объекта сохраняется в модуле накопления результатов моделирования. Взаимодействие с хранилищем осуществляется через объектный сервис доступа к БД.

Результаты множества сеансов моделирования с одинаковыми начальными данными могут быть обработаны статистическими методами с целью выявления усредненных моделей поведения различных параметров для определенных классов нештатных и аварийных ситуаций. Эта функция поддерживается сервисом статистической обработки и отображения результатов моделирования. Результаты статистической обработки в дальнейшем могут использоваться для автоматической классификации типов нештатных и аварийных ситуаций и поддержки принятия решений.

Реализация ПК. Комплекс разрабатывается для работы в среде ОС MS BC 3.0 и выше. Однако использование при разработке платформы Java [5] позволяет эксплуатировать комплекс в других операционных средах. Клиентская часть реализована



Рис. 3

вана с использованием библиотеки Swing, серверные службы – на основе контейнера сервлетов Apache Tomcat 6.0 и библиотеки JAX-WS 2.2.7.

Разработанный ПК СИМ предназначен для использования в качестве тренажера специалистов-анализаторов результатов испытательных пусков с целью повышения уровня их практической подготовки.

Возможным направлением дальнейшего развития ПК СИМ является расширение структурной модели ракет за счет добавления аналитических и имитационных моделей функционирования различных агрегатов и систем изделий, что позволит оценивать их влияние на динамику полета и расширить класс моделируемых нештатных и аварийных ситуаций.

СПИСОК ЛИТЕРАТУРЫ

1. Лысенко Л. Н. Наведение и навигация баллистических ракет: Учеб. пособие. М.: Изд-во МГТУ им. Н. Э. Баумана, 2007.
2. Сихарулидзе Ю. Г. Баллистика и наведение летательных аппаратов. М.: БИНОМ. Лаборатория знаний, 2011.
3. Дмитриевский А. А., Лысенко Л. Н. Внешняя баллистика: Учеб. для студентов вузов. М.: Машиностроение, 1991.
4. Машимов М. М. Геодезия. Теоретическая геодезия: Справ. пособие. М.: Недра, 1991.
5. Эккель Б. Философия Java. СПб.: Питер, 2009.

M. G. Panteleyev, A. V. Rumin, S. V. Lebedev

SIMULATION SOFTWARE FOR TRAINING OF SPECIALISTS-ANALYZERS IN THE FIELD OF ROCKET TECHNOLOGY

The problems in creating simulation software for training of specialists-analyzers of results in test launches are discussed. Mathematical models of missiles used in the development of the program and the method of modeling abnormal and emergency situations are considered. The software architecture of the complex is presented.

Simulation of test launches, mathematical model, software complex, architecture



СВЕДЕНИЯ ОБ АВТОРАХ

Беляев Сергей Алексеевич

Доцент кафедры математического обеспечения и применения ЭВМ СПбГЭТУ, канд. техн. наук.

E-mail: bserge@bk.ru.

Васильев Александр Владимирович

Заместитель главного конструктора по направлению ОАО «НИЦ СПб ЭТУ», канд. техн. наук.

Тел.: (812) 703-75-86 (доб. 177).

E-mail: vasiljev@nicetu.spb.ru.

Герцман Владимир Евгеньевич

Главный конструктор проекта ОАО «НИЦ СПб ЭТУ», канд. техн. наук.

Тел.: (812) 313-64-74.

E-mail: hertz@nicetu.spb.ru.

Голяк Сергей Сергеевич

Старший менеджер группы информационной безопасности Службы безопасности ОАО «Магнитогорский металлургический комбинат».

Тел.: +7 (908) 086-09-00.

E-mail: golyak.ss@mmk.ru.

Гордеев Станислав Александрович

Заместитель директора – начальник отдела связи и инфраструктуры информационных технологий ООО «ММК-Информсервис».

Тел.: +7 (3519) 24-32-41.

E-mail: gordeev.sa@mmk.ru.

Горошков Игорь Александрович

Системный архитектор отдела проектирования ПО, ОАО «НИЦ СПб ЭТУ».

Тел.: (812) 313-64-69.

E-mail: igor.goroshkov@nicetu.spb.ru.

Жукова Наталья Александровна

Ассистент кафедры математического обеспечения и применения ЭВМ СПбГЭТУ, канд. техн. наук.

Тел.: (812) 234-25-03.

E-mail: nazhukova@mail.ru.

Зименков Игорь Владимирович

Старший системный архитектор ОАО «НИЦ СПб ЭТУ».

Тел.: +7 (812) 313-78-09.

E-mail: zimenkov.igor@nicetu.spb.ru.

Ивановский Сергей Алексеевич

Зав. кафедрой математического обеспечения и применения ЭВМ СПбГЭТУ, канд. техн. наук, доцент.

Тел.: +7 (921) 913-79-21.

E-mail: saivanovsky@mail.ru.

Калишенко Евгений Леонидович

Начальник группы ОАО «НИЦ СПб ЭТУ», магистр техники и технологии.

Тел.: +7 (911) 716-81-27.

E-mail: kel@nicetu.spb.ru.

Кирьянчиков Владимир Андреевич

Доцент кафедры математического обеспечения и применения ЭВМ СПбГЭТУ, канд. техн. наук, доцент.

Тел.: +7 (911) 280-80-00.

E-mail: vak@nicetu.spb.ru.

Королёв Юрий Алексеевич

Аспирант кафедры автоматизированных систем обработки информации и управления СПбГЭТУ.

Тел.: +7 (962) 704-31-54.

E-mail: yury.king@gmail.com.

Кулаков Сергей Александрович

Ведущий специалист отдела системного проектирования ОАО «НИЦ СПб ЭТУ».

Тел.: (812) 313-64-69.

E-mail: kulakov.sergey@nicetu.spb.ru.

Куприянов Михаил Степанович

Профессор кафедры вычислительной техники СПбГЭТУ, д-р техн. наук.

Тел.: (812) 952-06-42.

E-mail: mikhail.kupriyanov@gmail.com.

Лебедев Сергей Вячеславович

Аспирант кафедры вычислительной техники СПбГЭТУ.

Тел.: +7 (911) 907-57-70.

E-mail: stativv@rambler.ru.

Лисс Анна Александровна

Главный конструктор проекта ОАО «НИЦ СПб ЭТУ», канд. техн. наук, доцент.

Тел.: (812) 703-71-88.

E-mail: Anna@nicetu.spb.ru.

Литвиненко Наталья Максимовна

Старший системный аналитик ОАО «НИЦ СПб ЭТУ».

Тел.: +7 (911) 089-02-28.

E-mail: natalya.litvinenko@nicetu.spb.ru.

Орешко Николай Иванович

Ведущий научный сотрудник ОАО «НИЦ СПб ЭТУ», канд. техн. наук.

Тел.: +7 (921) 751-75-58.

E-mail: oreshko@nicetu.spb.ru.

Пантелеев Михаил Георгиевич

Доцент кафедры вычислительной техники СПбГЭТУ, канд. техн. наук.

Тел.: +7 (921) 312-89-28.

E-mail: MPanteleyev@gmail.com.

Разумовский Геннадий Васильевич

Доцент кафедры вычислительной техники СПбГЭТУ, канд. техн. наук.

Тел.: +7 (962) 680-15-83.

E-mail: razumovsky@nicetu.spb.ru.

Родионов Сергей Васильевич

Доцент кафедры вычислительной техники СПбГЭТУ, канд. техн. наук.

Тел.: +7 (965) 077-89-86.

E-mail: sv-rodion@mail.ru.

Романенко Сергей Александрович

Доцент кафедры математического обеспечения и применения ЭВМ СПбГЭТУ, канд. техн. наук.

Тел.: +7 (921) 312-03-22.

E-mail: s_romanenko@mail.ru.

Романенко Юлия Владимировна

Начальник отдела мультимедиа-приложений и обучающих систем ОАО «НИЦ СПб ЭТУ».

Тел. +7 (921) 900-54-59.

E-mail: yvrom@mail.ru.

Романов Аркадий Сергеевич

Аспирант кафедры математического обеспечения и применения ЭВМ СПбГЭТУ.

Тел.: +7 (952) 390-49-99.

E-mail: arkadyrom@gmail.com.

Рюмин Алексей Владимирович

Руководитель проекта ОАО «НИЦ СПб ЭТУ», канд. техн. наук.

Тел.: +7 (812) 373-78-20.

E-mail: ryumin.aleksey@nicetu.spb.ru.

Сигов Евгений Николаевич

Системный архитектор ОАО «НИЦ СПб ЭТУ».

Тел.: (812) 313-78-09.

E-mail: evgeniy.sigov@nicetu.spb.ru.

Скрипникова Александра Сергеевна

Начальник отдела ОАО «НИЦ СПб ЭТУ».

Тел.: +7 (965) 036-09-01.

E-mail: alexandra.skripnikova@gmail.com.

Соколов Игорь Сергеевич

Заместитель начальника научно-исследовательского отдела ОАО «НИЦ СПб ЭТУ».

Тел.: +7 (904) 619-48-30.

E-mail: igor.s.sokolov@gmail.com.

Суворов Андрей Анатольевич

Заместитель начальника департамента разработки по системной интеграции ОАО «НИЦ СПб ЭТУ».

Тел.: (812) 313-78-09.

E-mail: aas@nicetu.spb.ru.

Федорова Юлия Юрьевна

Инженер-тестировщик I категории ОАО «НИЦ СПб ЭТУ».

E-mail: fedorova.yulia@nicetu.spb.ru.

Фирсов Михаил Александрович

Аспирант, магистр по направлению «Программная инженерия СПбГЭТУ».

Тел.: +7 (905) 260-33-52.

E-mail: pvptm@mail.ru.

Холод Иван Иванович

Доцент кафедры вычислительной техники СПбГЭТУ, канд. техн. наук.

Тел.: +7 (921) 795-42-58.

E-mail: iiholod@mail.ru.

Черная Ольга Сергеевна

Начальник отдела тестирования ОАО «НИЦ СПб ЭТУ».

E-mail: chernaya.olga@nicetu.spb.ru.

Шестопалов Михаил Юрьевич

Проректор по научной работе СПбГЭТУ.

Тел.: +7 (921) 946-01-18.

E-mail: shestopalov_08@mail.ru.

Шичкина Юлия Александровна

Зав. кафедрой ФГБОУ ВПО «БрГУ».

Тел.: 8 (3953) 32-53-73.

E-mail: strage.y@mail.ru

Экало Александр Владимирович

Директор ФГУП «НИИ ПС», д-р техн. наук,
доцент.

Тел.: (812) 297-16-49.

E-mail: niipc@mail.ru.



В редакционный совет журнала «Известия СПбГЭТУ "ЛЭТИ"» необходимо представить:

• файлы на CD-диске (в отдельных случаях по предварительному согласованию с редсоветом допускается передача материалов по электронной почте):

- рукопись статьи;
- сведения об авторе(ах);

• документы на листах формата А4 (1 экз.):

- рукопись статьи (статья должна быть объемом 6 маш. с., обзорная статья – до 12 маш. с.);
- сведения об авторе(ах);
- рекомендация кафедры с указанием наименования тематического раздела, в котором предполагается опубликование материала;
- экспертное заключение о возможности опубликования материала в открытой печати;
- сопроводительное письмо (только для авторов, не работающих (не обучающихся) в СПбГЭТУ «ЛЭТИ»).

Сведения об авторах должны содержать:

1. Фамилию, имя, отчество (полностью) автора.
2. Основное место работы.
3. Должность.
4. Ученую степень, ученое звание.
5. Контактный телефон, адрес электронной почты.

Все сведения печатаются в строку (без пунктов) на русском и английском языках.

Правила оформления текста статьи

Статья должна быть тщательно отредактирована, содержать признаки научной публикации:

- текстовый редактор Microsoft Word версии не ниже 97;
- формат бумаги А4;
- параметры страницы: поля – верхнее, нижнее и левое 2,5 см, правое 2 см, верхний колонтитул 2 см, нижний колонтитул 2 см;

• основной текст статьи: шрифт «Times New Roman» 10,5 pt; выравнивание по ширине; первая строка с абзацным отступом 0,6 см; межстрочный интервал «Множитель 1.1»; автоматическая расстановка переносов.

На первой странице указывается индекс УДК (в левом верхнем углу; шрифт «Times New Roman» 14 pt; межстрочный интервал одинарный); далее даются инициалы и фамилии авторов, затем название статьи, аннотация статьи и ключевые слова на русском языке. Инициалы и фамилии авторов, название статьи, аннотация статьи и ключевые слова на английском языке размещаются в конце публикации после библиографического списка.

Перечень авторов разделен запятыми, инициалы перед фамилиями (шрифт «Times New Roman» 11,5 pt, обычный; выравнивание по правому краю, абзацный отступ слева 0,6 см; автоматические переносы отключены; интервалы перед абзацем 20 pt, после 8 pt, межстрочный интервал одинарный).

Название статьи набирается прописными буквами (шрифт «Times New Roman» 15 pt; жирный; выравнивание по левому краю; автоматические переносы отключены; интервалы перед абзацем 12 pt, после него – 12 pt; межстрочный интервал одинарный).

Аннотация содержит 3–7 строк, характеризующих содержание статьи (шрифт «Times New Roman» 8,5 pt, курсив; выравнивание по ширине; отступы: слева и справа 0,6 см; интервалы перед абзацем 6 pt, после 0 pt, межстрочный интервал одинарный).

Ключевые слова состоят из 3–10 слов и словосочетаний, отражающих содержание статьи, разделенных запятыми, в конце их перечисления точка не ставится; шрифт «Times New Roman» 9,5 pt, жирный; выравнивание по левому краю; автоматические переносы отключены; интервалы перед абзацем 12 pt, после него 12 pt; межстрочный интервал одинарный).

Далее следует текст статьи.

После основного текста следует библиографический список, который включает:

- *заголовок* «Список литературы» набирается прописными буквами (шрифт «Times New Roman» 11 pt, жирный; выравнивание по центру; интервалы: перед абзацем 12 pt, после него 8 pt; межстрочный интервал «Одинарный»);

➤ *библиографическое описание источника* с порядковым номером ссылки на него по тексту, начиная с первого, выполненное по ГОСТ 7.3–2003. Библиографическое описание документа. Введ. 01.07.2004 / Госкомстандарт. М., 2004 (отдельным абзацем; шрифт «Times New Roman» 8,5 pt; первая строка с абзачным отступом 0,6 см; выравнивание по ширине; межстрочный интервал одинарный).

Библиографический список не должен превышать 10 (для обзорных заказных статей – 20) наименований: приводятся только источники, на которые есть ссылки в тексте (ссылки на неопубликованные и нетиражированные работы не допускаются).

Для создания *формул* используется редактор MathType. Необходимо использовать следующие установки редактора формул. Размеры: «полный» 10,5 pt, «подстрочный» 9 pt, «под-подстрочный» 7 pt, «символ» 14,5 pt, «подсимвол» 12,5 pt; стили: текст, функция, число – шрифт «Times New Roman», вектор-матрица – шрифт «Times New Roman», жирный; греческий малый, греческий большой, символ – шрифт «Symbol», прямой; переменная – шрифт «Times New Roman», курсив.

При наборе формул без использования редактора формул индексы ставятся размером 9 pt со смещением на 3 pt вверх или вниз (Главная → Шрифт → Дополнительно → Смещение) в зависимости от того, надстрочный или подстрочный индекс. Не использовать Шрифт → Видоизменение → надстрочный/подстрочный. Если у величины есть одновременно и надстрочные, и подстрочные символы, то формула создается только в редакторе MathType.

Формулы, помещенные на отдельной строке, набираются с интервалами: перед формулой 4 pt, после нее 4 pt; межстрочный интервал одинарный.

На каждый рисунок и таблицу в тексте статьи необходимо дать ссылку. Каждый рисунок и таблица должны быть пронумерованы (напр.: рис. 1, табл. 3).

Текст в *таблицах* печатается через одинарный интервал, шрифт «Times New Roman»; основной текст 9 pt, индексы 7 pt, подындекс 5,5 pt. Нумерационный заголовок содержит слово «Таблица» и ее номер арабскими цифрами (без знака номера перед ними и без точки на конце), интервал после 4 pt; выравнивается по правому краю таблицы и выделяется светлым курсивом; интервалы: перед таблице 6 pt, после нее 6 pt. Ширина таблицы не более 7,9 см, большие таблицы – не более 16,5 см.

Рисунки в электронном виде и подписи к ним создаются средствами Word или другими программами (CorelDRAW, Visio, Adobe Illustrator, Excel (с предоставлением оригинала рисунка в электронном виде)) в черно-белом виде. Качество рисунков и фотографий (в форматах .jpeg, .tiff) должно быть не менее 300 dpi. Основные линии на рисунках (границы блоков и соединительные линии на схемах, линии графиков) имеют толщину 1 pt, вспомогательные (линии-выноски, оси, размерные линии) – 0,6 pt. Не допускается вставлять рисунки в рамки или полотно.

Описание содержания рисунка, а также введенных на нем обозначений следует приводить в основном тексте статьи. Подпись под рисунком содержит только его номер (шрифт «Times New Roman», курсив, 9 pt) и располагается по центру под рисунком (например, *Рис. 1*). Все тексты и обозначения на рисунке даются шрифтом размером 9 pt, индексы 7 pt, подындексы 5,5 pt с одинарным интервалом; цифровые обозначения на линиях выносок и номер рисунка даются курсивом. Буквенные обозначения фрагментов рисунка (шрифт «Times New Roman», курсив, 9 pt) ставятся под фрагментом перед нумерационным заголовком; в тексте ссылка на фрагмент ставится после нумерационного заголовка через запятую (например, рис. 1, *a*); интервалы: перед и после рисунка 6 pt. Ширина рисунка не более 7,9 см, большие рисунки – не более 16,5 см.

Перечень основных тематических направлений журнала:

- Радиоэлектроника и телекоммуникации.
- Физика твердого тела и электроника.
- Информатика, управление и компьютерные технологии.
- Автоматизация и управление.
- Электротехника.
- Приборостроение и информационные технологии.
- Биотехнические системы в медицине и экологии.
- Управление качеством, инновационный и антикризисный менеджмент.
- Гуманитарные науки.
- История науки, образования и техники.
- Современные технологии в образовании.

Рукописи аспирантов печатаются бесплатно.

Технические вопросы можно выяснить по адресу: Izvestiya-leti@yandex.ru

Издательство СПбГЭТУ «ЛЭТИ» приносит свои извинения за допущенную в «Известиях СПбГЭТУ "ЛЭТИ"» вып. 8 за 2013 г. ошибку:

- на с. 8 напечатано: значение эффективной дозы E [мкЗв/ч];
следует читать: значение эффективной дозы E [мкЗв].