СИСТЕМЫ И СРЕДСТВА ИНФОРМАТИКИ

Научный журнал Российской академии наук
(издается под руководством Отделения нанотехнологий и информационных технологий РАН)

Издается с 1989 года
Журнал выходит ежеквартально

Учредители:
Российская академия наук
Институт проблем информатики Российской академии наук

Журнал входит в систему Российского индекса научного цитирования (РИНЦ):
http://elibrary.ru/title_about.asp?id=28980
Журнал включен в базу данных CrossRef (систему DOI — Digital Object Identifier),
в базу данных Ulrich's periodicals directory
и в информационную систему «Общероссийский математический портал Math-Net.Ru»

Журнал реферируется в «Реферативном журнале» ВИНИТИ
и в системе Google Scholar

Журнал «Системы и средства информатики»
включен в «Перечень российских рецензируемых журналов,
в которых должны быть опубликованы основные научные результаты диссертаций
на соискание ученых степеней доктора и кандидата наук», утвержденный ВАК

http://www.ipiran.ru/journal/collected

СИСТЕМЫ И СРЕДСТВА ИНФОРМАТИКИ

Том 24 № 2 Год 2014

СОДЕРЖАНИЕ

СИСТЕМЫ И СРЕДСТВА ИНФОРМАТИКИ

Том 24 № 2 Год 2014

СОДЕРЖАНИЕ

МЕТОДЫ И СРЕДСТВА ОПТИМАЛЬНОГО ПЛАНИРОВАНИЯ ПАРАМЕТРОВ ПРОЦЕССОВ В СИСТЕМАХ ПОСЛЕПРОДАЖНОГО ОБСЛУЖИВАНИЯ ИЗДЕЛИЙ НАУКОЕМКОЙ ПРОДУКЦИИ*

***И. Н. Синицын*[1], *А. С. Шаламов*[2], *И. В. Сергеев*[3], *Э. Р. Корепанов*[4], *В. В. Белоусов*[5], *Т. С. Гумникова*[6], *В. С. Шоргин*[7], *Е. С. Агафонов*[8]**

Аннотация: Рассматриваются методы и инструментальные программные средства оптимизации систем послепродажного обслуживания (СППО) на основе стоимостных критериев для заданного уровня коэффициента технической исправности. Дается краткий обзор современных подходов к управлению жизненным циклом (ЖЦ) изделий наукоемкой продукции (ИНП), основанных на западной концепции CALS (Continuous Acquisition and Life cycle Support) — непрерывной информационной поддержки поставок и жизненного цикла и ее российском аналоге — ИПИ (интегрированной информационной поддержке изделий). Представлен новый стохастический подход к моделированию СППО ИНП. Данный подход использован при решении актуальной задачи оптимального проектирования и эксплуатации системы обслуживания по стоимостным критериям. Описываются методы поиска оптимальных значений параметров систем обслуживания с учетом различных критериев эффективности. Демонстрируется методическое и инструментальное программное обеспечение оптимизации затрат на поставки запасных частей (ЗЧ) и ремонт на годы вперед, вплоть до списания изделия, в условиях ограничений на выделяемые финансовые ресурсы и необходимости поддерживать заданный уровень коэффициента технической исправности.

Ключевые слова: система послепродажного обслуживания; изделие наукоемкой продукции; ремонт и поставки запасных частей; уровень готовности (исправности) парка изделий; стоимость ремонта и поставок; ограниченный бюджет; оптимизация программ ремонта и поставок запасных частей; инструментальное программное обеспечение

DOI: 10.14357/08696527140201

*Работа выполнена при частичной финансовой поддержке Программы ОНИТ РАН «Интеллектуальные информационные технологии, системный анализ и автоматизация» (проект 1.7).

[1]Институт проблем информатики Российской академии наук, sinitsin@dol.ru

[2]Институт проблем информатики Российской академии наук, a-shal5@yandex.ru

[3]Институт проблем информатики Российской академии наук, ISergeev@ipiran.ru

[4]Институт проблем информатики Российской академии наук, Ekorepanov@ipiran.ru

[5]Институт проблем информатики Российской академии наук, VBelousov@ipiran.ru

[6]ОАО «Рособоронэкспорт», klimtat50@yandex.ru

[7]Институт проблем информатики Российской академии наук, VShorgin@ipiran.ru

[8]Институт проблем информатики Российской академии наук, EAgafonov@ipiran.ru

1 Введение

Современные подходы к использованию информационно-вычислительных систем в сфере управления стоимостью ЖЦ ИНП основаны на концепции CALS или ИПИ [1–3]. Разрабатываемые крупные СППО используют международные стандарты, а также методы интегрированной логистической поддержки (ИЛП). С этой целью создаются специализированные технико-экономические информационные модели ЖЦ ИНП, представляющие собой нормативные базы данных (БД) и технической документации. Эти модели носят статический характер, что не позволяет оперативно управлять качеством и стоимостью на заданном периоде эксплуатации изделий. Поэтому дополнительно используют БД мониторинга основных эксплуатационных и других необходимых характеристик изделия, позволяющих вычислять главные показатели эффективности управления, включая стоимость и технологичность эксплуатации.

Основные принципы ИЛП изложены в международных стандартах MIL STD 1388 (США) и Def Stan 00-600 “Integrated Logistic Support. Requirements for MOD Projects”, JSP-886 “The defense logistic support chain manual. Volume 7: Integrated logistic support” и др.

Интегрированная логистическая поддержка представляет собой организационно-технологический комплекс послепродажного обслуживания, обладающий современной системой управления логистическими процессами на основе информационно-телекоммуникационных сетей и новых технологий информационной и интеллектуальной поддержки принимаемых решений.

Особенностью системы ИЛП является то, что ее построение начинается одновременно с созданием ИНП и идет параллельно с ним до стадии эксплуатации. Далее на стадии эксплуатации система ИЛП обеспечивает эффективное применение ИНП, а на стадии утилизации — управление переработкой ИНП для вторичного использования материалов, входящих в состав изделий, с целью их максимальной реализации.

Однако при всем совершенстве и многообразии технологий ИЛП, предлагаемых этими стандартами, фактически остается нерешенной главная проблема — проблема оптимального проектирования системы обслуживания, ремонта и снабжения для обеспечения эксплуатации изделий с минимальной стоимостью на годы вперед, вплоть до их списания.

В настоящей работе представлены новые научные подходы к решению сформулированной проблемы на основе развития результатов [1, 4–21]. Они позволяют кардинально реформировать традиционные системы управления при создании и эксплуатации ИНП путем внедрения методов прогнозирования и оптимального планирования процессов расходования временны́х, материальных, трудовых и других ресурсов по критериям экономической целесообразности и эффективности.

2 Критерии эффективности систем послепродажного обслуживания

Как известно [3, 15], статистический анализ и параметрический синтез СППО на основе вероятностных критериев эффективности используются как на этапе проектирования СППО, так и в процессе функционирования.

К задачам на этапе проектирования СППО относятся:

- сравнение различных производственных и организационно-штатных структур;
- определение оптимальных параметров в выбранной структуре;
- сравнение различных политик пополнения ресурсами;
- выбор оптимальных параметров принятой политики пополнения;
- определение требований к структуре и параметрам систем, обеспечивающих функционирование СППО.

Методами статистического анализа в процессе функционирования СППО решаются следующие функциональные задачи:

- обоснование оптимального (рационального) перераспределения материальных и людских ресурсов;
- прогнозирование состояния СППО на заданном промежутке времени;
- выработка рекомендаций по оптимальному расходованию ресурсов;
- уточнение политики пополнения ресурсов и ее параметров.

Под вероятностным критерием эффективности системы понимают вероятностную оценку ее функционала качества.

Определим функционал качества СППО в виде скалярного соотношения

$$\mathcal{J} = \mathcal{J}(Y, \Lambda)\,,$$

где $Y \in R^n$ — вектор фазовых координат СППО; $\Lambda \in R^{n1}$ — вектор параметров системы. Проектируя СППО или уточняя ее структуру в процессе функционирования, необходимо стремиться к тому, чтобы выполнялось соотношение

$$\mathcal{J}(Y, \Lambda) \to \min_{\Lambda}\,. \tag{1}$$

Для случайного вектора $Y(t)$ решение задачи (1) дает функциональную зависимость $\Lambda = \widehat{\Lambda}(Y)$ со случайным аргументом. Поэтому часто пользуются другим критерием:

$$\mathcal{J}_1 = M\left[\mathcal{J}(Y, \Lambda)\right] \to \min_{\Lambda}\,, \tag{2}$$

т. е. минимизируют математическое ожидание величины $\mathcal{J}(Y, \Lambda)$.

Иногда определенную роль играет дисперсия критерия. В этом случае вполне допустимо несколько поступиться значением математического ожидания, чтобы уменьшить возможный разброс результатов, т. е. уменьшить значение дисперсии значений критерия:

$$\mathcal{J}_2 = M\left[(\mathcal{J}(Y,\Lambda) - \mathcal{J}_1)^2\right] \to \min_{\Lambda}.$$

Может возникнуть ситуация, когда при выбранном векторе Λ слишком неудовлетворительным является значение критерия (2), поэтому используется комбинированный критерий — второй начальный момент величины (1):

$$\mathcal{J}_3 = \mathcal{J}_1^2 + \mathcal{J}_2 \to \min_{\Lambda}. \tag{3}$$

Результаты использования такого вида критерия наиболее плодотворны, когда слагаемые в (3) имеют взаимно противоречивый характер.

Обобщенным критерием эффективности СППО является следующий критерий:

$$\mathcal{J}_4 = P(\Lambda) \to \max_{\Lambda},$$

где $P(\Lambda)$ — вероятность выполнения задачи, определяемой функциональным назначением СППО.

Вероятность $P(\Lambda)$ определяется как интеграл от плотности вероятности $f_y(y;\Lambda)$ вектора фазовых координат Y системы по области, соответствующей успешному решению функциональной задачи.

Приведенные выше варианты критериев качества позволяют решать задачи, связанные с проектированием СППО. При этом функционирование системы рассматривается на некотором промежутке времени $[0,T]$, где T — достаточно большой период. В условиях стохастической неопределенности некоторых заранее не известных факторов, влияющих на работу СППО, моделировать политику изменения параметров Λ в виде функций времени, как правило, невозможно, поэтому их полагают некоторыми константами.

Значения критериев эффективности вычисляются на момент времени $t = T$ (такие критерии называют терминальными). При этом стремятся к тому, чтобы постановка задачи на оптимизацию вектора параметров Λ оказывалась корректной, ибо в противном случае в конкретной ситуации эффективность системы может быть резко снижена за счет несоответствия заранее выбранных параметров тем условиям, в которые попадает СППО. Корректность постановки задачи позволяет иметь достаточный ресурс возможностей по маневрированию силами и средствами в приемлемые сроки с целью выравнивания ситуации (осуществления ситуационного управления) без существенного снижения эффективности СППО.

При уточнении параметров Λ в процессе функционирования СППО на некотором сравнительно небольшом промежутке времени $[t_1,t_2]$ при решении функциональных задач возможно достаточно точное его описание с помощью

вероятностной модели. В результате вектор Λ может оказаться более адекватным реальной обстановке, чем это предусмотрено проектом системы. На практике такое уточнение реализуется с помощью временны́х изменений организационно-штатных и производственных структур и связей в СППО. Это уточнение может быть осуществлено либо в виде нового значения вектора Λ на всем периоде $[t_1, t_2]$, либо в виде нескольких его значений, отличающихся на различных промежутках времени внутри отрезка $[t_1, t_2]$, либо в виде функции времени $\Lambda(t)$.

В последнем случае решение задачи по определению наилучшего вида зависимости $\Lambda(t)$ приводит к оптимизации в функциональном пространстве и может трактоваться при наличии математической модели системы как оптимальное функциональное управление.

В классической теории управления ресурсами [22] задача проектирования систем типа «склад» решается на основе статистического анализа установившихся процессов по критерию $\mathcal{J}_1$ (2). Это обусловлено тем, что процессы в СППО, в частности на складе, имеют непрерывно-дискретное время (непрерывный спрос и дискретные поставки для пополнения). А поскольку теория таких процессов до сих пор разработана недостаточно и существуют лишь подходы к решению задач в установившихся режимах, то возможности учета динамики нестационарных элементов в традиционных системах поддержки поставок отсутствуют.

Решение упрощенной задачи оптимизации по одному критерию изучено в [10, 11], в данной работе рассматривается случай многокритериальной совместной оптимизации параметров СППО.

В качестве базового рассмотрим случай с периодической подачей заявок на пополнение. Пусть компоненты критерия эффективности J системы отражают затраты на хранение ресурсов, на пополнение, штрафы за недопоставки потребителям, доходы от использования ресурсов в некоторых технологических процессах, затраты на создание дополнительных систем обслуживания ресурсов при их восстановлении и подготовке к повторному использованию и т. д.

Путем расширения фазового пространства СППО значение критерия эффективности (вероятностные характеристики) можно получить в любой текущий момент времени $t > t_0$. Если критерий $J(Y, \Lambda)$ относится к терминальному типу, то его значение имеет полезный смысл только на конце отрезка $[0, T]$.

Вероятностные характеристики фазового вектора $Y(t)$ системы описываются дифференциальными уравнениями, приведенными в [3, 12]. Вектор пополнения $\Pi(t_j)$ является прогнозируемой величиной, неслучайным векторным параметром, подбираемым в процессе оптимизации всего вектора Λ.

Рассмотрим способ решения задачи оптимизации параметров Λ по критерию (1).

Пусть требуется найти такую совокупность составляющих вектора Λ, при которой обеспечивается $\max J_1$. Ограничимся рассмотрением случая гауссовского распределения СтП $Y(t)$. Тогда критерий оптимальности является функцией совокупности параметров

$$J_1 = G(m, \theta, \Lambda)\,,$$

где m и θ — вектор математического ожидания и ковариационная матрица вектора Y; G — скалярная функция, априори не известная; Λ имеет размерность $n_1 \times 1$. В общем случае на аргумент Λ функции G могут быть наложены ограничения, записываемые в виде соотношения:

$$M(\Lambda) = 0\,, \tag{4}$$

где $M(\cdot)$ — в общем случае векторная функция размерности $n_2 \times 1$. Будем предполагать, что функции G и M являются дифференцируемыми по аргументу Λ, причем производные непрерывны.

Применим далее для определения оптимальной совокупности Λ градиентный метод. С этой целью введем сначала функцию Лагранжа

$$L(\Lambda) = G(\Lambda) + \Psi^{\mathrm{T}} M(\Lambda)\,, \tag{5}$$

где Ψ — векторный коэффициент Лагранжа размерности $n_2 \times 1$. Из необходимых условий существования стационарной точки следует, что

$$\frac{d\,\mathcal{L}}{d\Lambda} = 0 \quad \text{или} \quad \frac{dG}{d\Lambda} + \left[\frac{d\,\mathcal{M}}{d\Lambda}\right]^{\mathrm{T}} \Psi = 0\,.$$

Если ограничение (4) удовлетворено, то справедливо приближенное равенство

$$\Delta J_1 \cong \left[\frac{d\,\mathcal{L}}{d\Lambda}\right]^{\mathrm{T}} \Delta\Lambda\,,$$

где $\Delta\Lambda$ и ΔJ_1 — приращение вектора параметров Λ и соответствующее ему приращение критерия оптимальности J_1.

Чтобы максимизировать ΔJ_1, необходимо вычислить градиент $d\,\mathcal{L}/d\Lambda$ и найти новое значение J_1, изменяя $\Delta\Lambda$ в направлении градиента. Поэтому запишем

$$\Delta\Lambda = \xi \frac{d\,\mathcal{L}}{d\Lambda}\,. \tag{6}$$

Отсюда следует, что при определении оптимума приращения критерия оптимальности J_1, соответствующие наискорейшему подъему (спуску) в сторону экстремума, можно вычислять по формуле:

$$\Delta J_1 \cong \pm\xi \left[\frac{d\,\mathcal{L}}{d\Lambda}\right]^{\mathrm{T}} \left[\frac{d\,\mathcal{L}}{d\Lambda}\right]\,,$$

где ξ — некоторый коэффициент пропорциональности, подбираемый из условия максимальной скорости подъема (спуска).

Задача определения оптимального вектора $\tilde{\Lambda}$ решается следующим методом итераций. Пусть некоторое значение $\Lambda^{(i)}$ вектора Λ удовлетворяет требованию (4). Найдем удовлетворяющее этим ограничениям значение $\Lambda^{(i+1)}$, при котором $\Lambda^{(i+1)} - \Lambda^{(i)} = \Delta\Lambda$, а разность $[G(\Lambda^{(i+1)} - G(\Lambda^{(i)}]$ максимальна.

Рассмотрим выражение (4). Поскольку оно должно удовлетворяться и для $\Lambda^{(i)}$, и для $\Lambda^{(i+1)}$, то отсюда следует, что должно выполняться условие

$$\Delta M \cong \frac{d\,\mathcal{M}}{d\Lambda}\,\Delta\Lambda = 0\,.$$

Подставим сюда выражение (6). С учетом (5) получим

$$\frac{d\,\mathcal{M}}{d\Lambda}\left[\frac{dG}{d\Lambda} + \left(\frac{d\,\mathcal{M}}{d\Lambda}\right)^{\mathrm{T}}\Psi\right] = 0\,. \tag{7}$$

После подстановки заданного значения $\Lambda^{(i)}$ под знаки соответствующих функций в (7) получаем $\Psi^{(i)}$:

$$\Psi^{(i)} = -\left[\frac{d\,\mathcal{M}\left(\Lambda^{(i)}\right)}{d\Lambda^{(i)}}\left(\frac{d\,\mathcal{M}\left(\Lambda^{(i)}\right)}{d\Lambda^{(i)}}\right)^{\mathrm{T}}\right]^{-1}\frac{d\,\mathcal{M}\left(\Lambda^{(i)}\right)}{d\Lambda^{(i)}}\,\frac{dG\left(\Lambda^{(i)}\right)}{d\Lambda^{(i)}}\,. \tag{8}$$

Далее после нахождения $\Psi^{(i)}$ определяется градиент функции Лагранжа:

$$\frac{d\,\mathcal{L}\left(\Lambda^{(i)}\right)}{d\Lambda^{(i)}} = \frac{dG\left(\Lambda^{(i)}\right)}{d\Lambda^{(i)}} + \left(\frac{d\,\mathcal{M}\left(\Lambda^{(i)}\right)}{d\Lambda^{(i)}}\right)^{\mathrm{T}}\Psi^{(i)}\,.$$

Наконец, в соответствии с (6) получаем

$$\Lambda^{(i+1)} = \Lambda^{(i)} + \xi\frac{d\mathcal{L}\left(\Lambda^{(i)}\right)}{d\Lambda^{(i)}}\,. \tag{9}$$

Итерационная процедура останавливается при выполнении условия

$$G\left(\Lambda^{(i+1)}\right) - G\left(\Lambda^{(i)}\right) < \varepsilon\,, \tag{10}$$

где ε — заранее заданная допустимая величина погрешности определения оптимума.

Метод позволяет получить решение, причем достаточно быстро, при условии, что критерий оптимальности является выпуклым относительно Λ. Это требует дополнительных исследований свойств критериальной функции $G(\Lambda)$. Обычно функция $G(\Lambda)$ априори не известна и может быть лишь определена как табулированная зависимость от Λ. Это связано с тем, что значения функции $G(\Lambda)$ определяются в результате интегрирования уравнений для первых двух вероятностных моментов при заданной совокупности параметров Λ. Поэтому она может быть представлена как аппроксимация (огибающая) дискретного ряда значений $G(\Lambda^{(i)})$. Значит, вычисление градиентов $d\mathcal{L}/d\Lambda$ в точках $\Lambda^{(i)}$ требует использования эффективных численных методов приближения функций [23]. Приведенная итерационная процедура обладает недостаточно высокой сходимостью. Требуется коррекция результатов, получаемых по формуле (9). Практические рекомендации, приведенные в [23], сводятся к использованию дополнительных вычислений по формуле:

$$\Lambda_*^{(i+1)} = \Lambda^{(i+1)} - \\ - \left(\frac{d\,\mathcal{M}\left(\Lambda^{(i)}\right)}{d\Lambda^{(i)}}\right)^{\mathrm{T}} \left[\left(\frac{d\,\mathcal{M}\left(\Lambda^{(i)}\right)}{d\Lambda^{(i)}}\right)\left(\frac{d\,\mathcal{M}\left(\Lambda^{(i)}\right)}{d\Lambda^{(i)}}\right)^{\mathrm{T}}\right]^{-1} M\left(\Lambda^{(i+1)}\right). \quad (11)$$

В этом случае итеративный процесс продолжается до тех пор, пока кроме (10) не будет достигнута достаточная близость $M(\Lambda^{(i+1)})$ к гиперповерхности $M(\Lambda) = 0$.

3 Оптимальное планирование параметров процессов в системах послепродажного обслуживания

В области прогнозирования процессов смешанной природы в СППО ИНП возможна различная постановка задач. Наиболее всеохватывающей является глобальная оптимизация комплексной системы «эксплуатация ИНП – обслуживание и ремонт – материально-техническое обеспечение» по критериям стоимости при ограничении, накладываемом на коэффициент (уровень) технической готовности. При этом подсистема эксплуатации ИНП является источником исходной информации (параметры парка ИНП, интенсивность использования по назначению и др.). В части подсистем обслуживания и ремонта оптимизации могут подлежать параметры производительности предприятий, оказывающих услуги по текущему обслуживанию и ремонту ИНП, а также по их капитальному ремонту. В подсистеме материально-технического обеспечения в первую очередь необходимо оптимизировать параметры поставок ЗЧ, комплектующих и оборудования (сроки, объемы), ремонта.

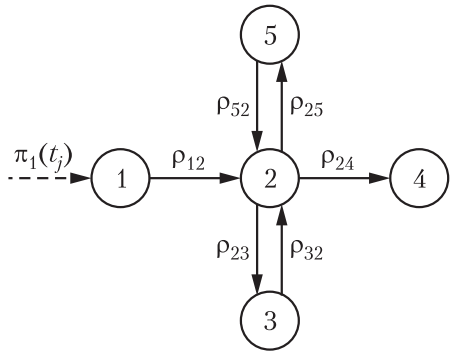


Рис. 1 Граф состояний парка изделий и их составных частей

Для демонстрации предлагаемых подходов ставится задача оптимизации затрат заказчика на поставки ЗЧ и ремонт на годы вперед, вплоть до списания ИНП в условиях ограничений на выделяемые финансовые ресурсы.

Эта задача имеет вполне самостоятельное значение в условиях, в некотором смысле диктуемых обстоятельствами, когда покупатель согласен с параметрами системы предоставления услуг по обслуживанию и ремонту, обусловленными понятными выгодами для поставщика.

Пусть состояние парка однотипных ИНП и их составных частей (СЧИ) описывается графом, представленным на рис. 1.

Вершины графа отображают возможные состояния составных частей изделия: 1 — исправные СЧИ как ЗЧ на складе; 2 — исправные СЧИ, эксплуатируемые в составе изделия; 3 — неисправные СЧИ, находящиеся на восстановительном ремонте у потребителя; 4 — СЧИ в состоянии списания; 5 — исправные СЧИ в составе ИНП, находящихся на профилактических работах.

В обороте находится определенное количество СЧИ каждого типа. В силу различных случайных факторов все они распределяются случайным образом между указанными выше состояниями.

Стрелками показаны дуги графа (см. рис. 1), описывающие переходы СЧИ и изделия в целом из одного состояния в другое. Обозначения дуг графа парами цифр kh отмечают переходы из состояния k в состояние h с интенсивностью ρ_{kh}. Состояния, в которых над СЧИ осуществляются какие-либо действия, соответствуют нахождению их в определенных технологических фазах, реализуемых системами массового обслуживания (СМО). Считается, что переходы k–h обусловлены случайными пуассоновскими потоками событий.

Периодические пополнения характеризуются периодом T_1 и объемом пополнения $\pi_1(t_i) = \pi_1(jT_1)$.

Введем в рассмотрение фазовый вектор системы $Y(t)$, составляющие которого $Y_i(t)$, $i = 1, \ldots, n$, суть количество СЧИ данного типа, находящихся в i-м состоянии.

Одним из важнейших показателей ИЛП является средний коэффициент технической готовности (исправности) на заданном промежутке времени $[0, T]$, который применительно к рассматриваемой системе определяется по формуле:

$$\overline{K}_u(T) = \frac{1}{T}\int\limits_0^T \frac{Y_2(\tau) + Y_5(\tau)}{N}\, d\tau\,,$$

где $Y_2(t)$, $Y_5(t)$ — количество исправных СЧИ; N — находящееся в эксплуатации количество СЧИ данного типа, строго согласованное с количеством эксплуатируемых ИНП (с учетом ИНП, находящихся в капитальном ремонте).

За критерий эффективности (оптимальности) примем выражение:

$$J = C_k \int\limits_0^T \left(\overline{K}_u - \overline{K}_{u\,\mathrm{тр}}\right)^2 d\tau\,,$$

где $\overline{K}_u(t)$ — среднее на интервале $[0, t)$ значение коэффициента технической готовности, случайная функция; $\overline{K}_{u\,\mathrm{тр}}$ — заданное заказчиком (требуемое) значение среднего на периоде $[0, T]$ коэффициента технической готовности парка ИНП; C_k — коэффициент.

Требуется получить оптимальные программы поставок ЗЧ и объемов ремонта на заданный период эксплуатации ИНП для обеспечения заданного уровня технической готовности изделий. При этом в системе действуют финансовые ограничения в виде годового бюджета.

В данном случае ограничение (4) связывает бюджет с расходами на закупки и ремонт и принимает вид:

$$C_{Rj}\lambda_{Rj} + C_{Pj}\lambda_{Pj} - C_j = 0\,,$$

где j — текущий год; C_{rj} — стоимость ремонта СЧИ; λ_{Rj} — производительность ремонта (СЧИ в год); C_{Pj} — стоимость закупки одной СЧИ; λ_{Pj} — объем поставки; C_j — годовой бюджет.

Алгоритм (8) и (9) вычисления оптимального значения $\tilde{\Lambda}$ вектора параметров с использованием градиентного метода в предположении, что критериальная функция $G(\Lambda)$ в момент $t = T$ является дифференцируемой по совокупности аргументов, определяется соотношениями:

$$\lambda_{Pj}^{(i+1)} = \lambda_{Pj}^{(i)} - \xi\left[\frac{\partial G}{\partial \lambda_{Pj}^{(i)}} - \frac{C_{Pj}}{C_{Pj}^2 + C_{Rj}^2}\left(C_{Pj}\frac{\partial G}{\partial \lambda_{Pj}^{(i)}} + C_{Rj}\frac{\partial G}{\partial \lambda_{Rj}^{(i)}}\right)\right];$$

$$\lambda_{Rj}^{(i+1)} = \lambda_{Rj}^{(i)} - \xi\left[\frac{\partial G}{\partial \lambda_{Rj}^{(i)}} - \frac{C_{Rj}}{C_{Pj}^2 + C_{Rj}^2}\left(C_{Pj}\frac{\partial G}{\partial \lambda_{Pj}^{(i)}} + C_{Rj}\frac{\partial G}{\partial \lambda_{Rj}^{(i)}}\right)\right].$$

Уточнение результатов по формуле (11) в каждой итерации получается с использованием следующих выражений:

$$\overline{\lambda}_{Pj}^{(i+1)} = \lambda_{Pj}^{(i)} - \xi_2 \frac{C_{Pj}}{C_{Pj}^2 + C_{Rj}^2} \left(C_{Rj}\lambda_{Rj} + C_{Pj}\lambda_{Pj} - C_j\right) ;$$

$$\overline{\lambda}_{Rj}^{(i+1)} = \lambda_{Rj}^{(i)} - \xi_2 \frac{C_{Rj}}{C_{Pj}^2 + C_{Rj}^2} \left(C_{Rj}\lambda_{Rj} + C_{Pj}\lambda_{Pj} - C_j\right) .$$

Значения частных производных $\partial G/\partial m_{10}^{(i)}$, $\partial G/\partial N_0^{(i)}$ и $\partial G/\partial \pi_1^{(i)}$ необходимо вычислять численным путем, используя аппроксимацию функции $G(m_1(0), N_0, \Lambda)$.

4 Инструментальный программный комплекс

В ИПИ РАН в 2013 г. создана версия 2.0 экспериментального программного комплекса прогнозирования и оптимизации «Оптимизация СППО».

Программы пополнений и ремонтов считаются оптимальными, если прогнозируемый при этих объемах уровень исправности на планируемом периоде эксплуатации ИНП минимально отличается от заданного с учетом действующих ограничений на объем годового бюджета и производительность ремонта СЧИ. Фактически это означает, что излишки ЗЧ на складе должны быть минимальными, что и подтверждается в численном эксперименте. Это равнозначно минимальной стоимости поставок.

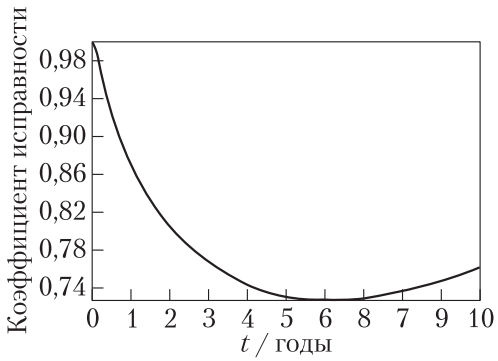


Рис. 2 Средний коэффициент исправности (без учета бюджета)

Далее продемонстрируем результаты работы программного комплекса прогнозирования и оптимизации в различных режимах.

Рисунки 2 и 3 соответствуют режиму поиска программ закупок ЗЧ и производительности ремонта по годам для поддержания заданного уровня коэффициента исправности 0,75 (см. рис. 2) без учета финансовых ограничений, которые определяются величиной годового бюджета.

В результате поиска были найдены следующие программы по годам. Для поставок ЗЧ: 0, 0, 1, 3, 4, 5, 6, 6, 6. Для производительности ремонта (СЧИ в год): 21, 23, 23, 20, 20, 20, 20, 21, 22. Общие затраты на закупку и ремонт в

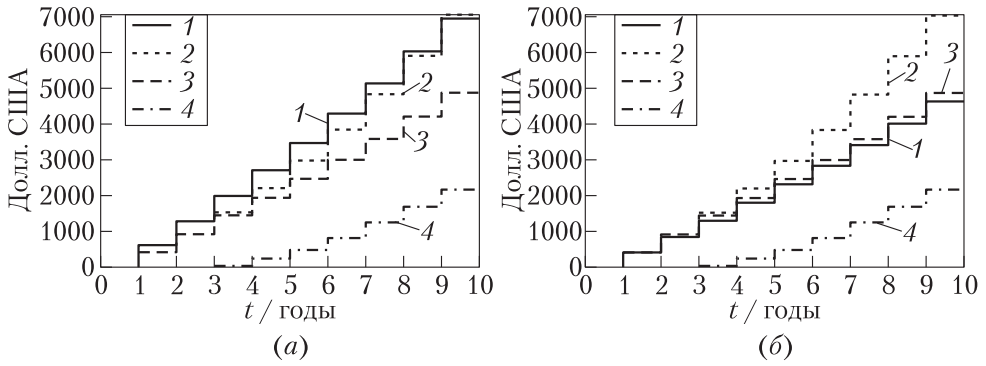


Рис. 3 Программы поставок и ремонта для высокого (*а*) и низкого (*б*) уровней бюджета: *1* — бюджет; *2* — общие затраты; *3* — стоимость ремонта; *4* — стоимость закупок

данном случае составят около 7150 долл. США. На рис. 3 показаны стоимости найденных программ вместе с уровнем бюджета. Рисунок 3, *а* соответствует случаю, когда затраты укладываются в выделенный бюджет, а рис. 3, *б* — случаю низкого уровня бюджета. Выполнение требований такого бюджета оказывается невозможным при удержании необходимого значения среднего коэффициента исправности.

На рис. 4 и 5 представлены результаты поиска программ поставок и ремонта с учетом финансовых ограничений. В данном случае общие затраты на закупку и ремонт не превышают величину выделенного бюджета на каждый год, но

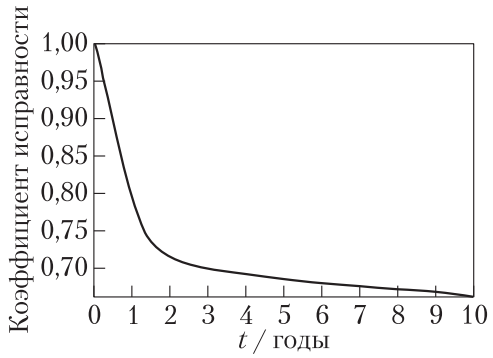


Рис. 4 Средний коэффициент исправности (с учетом низкого уровня бюджета)

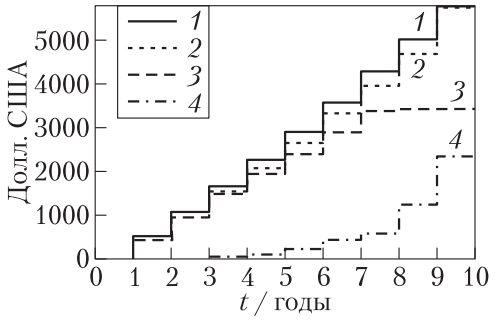


Рис. 5 Программы поставок и ремонта (с учетом низкого уровня бюджета): *1* — бюджет; *2* — общие затраты; *3* — стоимость ремонта; *4* — стоимость закупок

при этом не удается удерживать коэффициент исправности на уровне 0,75 (см. рис. 4).

На рис. 5 показаны программы, при которых выполняются условия бюджетных ограничений и поддерживается максимально возможный уровень коэффициента исправности. Были получены следующие значения программ по годам. Для поставок ЗЧ: 0, 0, 1, 1, 2, 3, 2, 9, 14. Для производительности ремонта (СЧИ в год): 21, 23, 23, 19, 18, 18, 17, 2, 0. Общая стоимость затрат составила примерно 5860 долл. США при суммарном бюджете в 5788 долл. США.

5 Заключение

Разработанное инструментальное программное обеспечение осуществляет автоматический поиск оптимальных программ поставок и ремонта по годам в условиях финансово-бюджетных ограничений. Оно позволяет эффективно решать задачу оптимизации не только для упрощенной системы, но и в случае сложных гибридных СППО.

В [17, 19] разработаны методы и алгоритмы оптимизации СППО на базе методов и средств канонических разложений.

Использование подобных комплексов прогнозирования и оптимизации изготовителями и заказчиками ИНП позволит им эффективно и оперативно решать задачи прогнозирования, оценки и управления процессами в СППО.

Литература

1. *Норенков И. П., Кузьмик П. К.* Информационная поддержка наукоемких изделий (CALS-технологии). — М.: МГТУ им. Н. Э. Баумана, 2002. 320 с.
2. *Wong K.* Стратегии PLM: удлинение ЖЦИ на крупных сервисно-ориентированных предприятиях // CAD/CAM/CAE Observer, 2008. № 2(38). С. 17–20.
3. *Синицын И. Н., Шаламов А. С.* Лекции по теории систем интегрированной логистической поддержки. — М.: ТОРУС ПРЕСС, 2012. 624 с.
4. *Пугачев В. С., Синицын И. Н.* Теория стохастических систем. — М.: Логос, 2000. 1000 с. (1-е изд.); 2004. 1000 с. (2-е изд.).
5. *Синицын И. Н.* Канонические представления случайных функций и их применение в задачах компьютерной поддержки научных исследований. — М.: ТОРУС ПРЕСС, 2009. 768 с.
6. *Синицын И. Н., Шаламов А. С.* Методологические аспекты современной интегрированной логистической поддержки изделий наукоемкой продукции // Системы высокой доступности, 2011. Т. 7. № 4. С. 48–74.
7. *Синицын И. Н., Шаламов А. С.* Проектирование CALS систем. Часть 1. Системы управления жизненным циклом изделий и их моделирование // Системы высокой доступности, 2012. Т. 8. № 3. С. 3–17.
8. *Синицын И. Н., Шаламов А. С.* Проектирование CALS систем. Часть 2. Аналитическое моделирование интегрированных систем послепродажного обслуживания

изделий наукоемкой продукции // Системы высокой доступности, 2012. Т. 8. № 4. С. 4–49.

9. *Синицын И. Н., Шаламов А. С., Сергеев И. В.* Проблемы моделирования и минимизации затрат на эксплуатацию изделий наукоемкой продукции на современном этапе // Кибернетика и высокие технологии XXI века (C&T-2012): Сб. докл. XIII Междунар. науч.-технич. конф. — Воронеж: Саквоее, 2012. Т. 2. С. 358–370.
10. *Синицын И. Н., Шаламов А. С., Синицын В. И.* Развитие систем интегрированной логистической поддержки изделий наукоемкой продукции // Оптико-электронные приборы и устройства в системах распознавания образов, обработки изображений и символьной информации (Распознавание-2012): Сб. мат-лов X Междунар. конф. — Курск: ЮЗГУ, 2012. С. 63–65.
11. *Синицын И. Н., Шаламов А. С., Сергеев И. В., Белоусов В. В., Агафонов Е. С.* Развитие средств интегрированной логистической поддержки изделий наукоемкой продукции на основе систем компьютерной математики // Системы компьютерной математики и их приложения: Сб. мат-лов XIII Междунар. науч. конф. — Смоленск: СмолГУ, 2012. Вып. 13. С. 119–124.
12. *Синицын И. Н., Шаламов А. С., Сергеев И. В., Синицын В. И., Корепанов Э. Р., Белоусов В. В., Агафонов Е. С., Шоргин В. С.* Методы и средства анализа и моделирования стохастических систем интегрированной логистической поддержки // Системы и средства информатики, 2012. Т. 22. № 2. С. 3–28.
13. *Синицын И. Н., Шаламов А. С., Синицын В. И., Агафонов Е. С.* Алгоритмическое и программное обеспечение обработки информации и синтеза систем интегрированной логистической поддержки // Оптико-электронные приборы и устройства в системах распознавания образов, обработки изображений и символьной информации (Распознавание-2013): Сб. мат-лов XI Междунар. науч.-технич. конф. — Курск: ЮЗГУ, 2013. С. 428–430.
14. *Синицын И. Н., Шаламов А. С.* Проектирование CALS систем. Часть 3. Аналитическое моделирование систем послепродажного обслуживания со смешанными потоками расходования, восстановления и пополнения запасов // Системы высокой доступности, 2013. Т. 9. № 1. С. 4–34.
15. *Синицын И. Н., Шаламов А. С.* Проектирование CALS систем. Часть 4. Статистический анализ и параметрический синтез систем послепродажного обслуживания // Системы высокой доступности, 2013. Т. 9. № 2. С. 4–35.
16. *Синицын И. Н., Шаламов А. С., Корепанов Э. Р., Белоусов В. В., Агафонов Е. С.* Инструментальная система автоматического поиска оптимальных программ поставок в системах послепродажного обслуживания изделий // Системы высокой доступности, 2013. Т. 9. № 2. С. 47–54.
17. *Синицын И. Н., Шаламов А. С., Кулешов А. А.* Нелинейное корреляционное моделирование и анализ надежности систем послепродажного обслуживания изделий наукоемкой продукции // Системы и средства информатики, 2013. Т. 23. № 1. С. 80–104.
18. *Синицын И. Н., Шаламов А. С., Корепанов Э. Р., Белоусов В. В., Сергеев И. В., Кулешов А. А.* Развитие алгоритмического и инструментального программного обеспечения для аналитического вероятностного моделирования и оптимизации процессов материально-технического обеспечения // Кибернетика и высокие технологии XXI века (C&T-2013): Сб. докл. XIV Междунар. науч.-технич. конф. — Воронеж: Саквоее, 2013. Т. 2. С. 375–384.

19. *Синицын И. Н., Шаламов А. С., Синицын В. И., Корепанов Э. Р., Белоусов В. В., Кулешов А. А.* Методы и средства оценки запасов и уровня готовности систем интегрированной логистической поддержки, основанные на канонических разложениях случайных функций // Современные проблемы прикладной математики, информатики, автоматизации, управления: Мат-лы 3-го Междунар. науч.-технич. семинара. — М.: ИПИ РАН, 2013. С. 115–126.
20. *Синицын И. Н., Шаламов А. С.* Моделирование и синтез системы послепродажного обслуживания продуктов на стороне поставщика // Системы высокой доступности, 2013. Т. 9. № 4. С. 12–24.
21. *Синицын И. Н., Шаламов А. С.* Моделирование и синтез системы послепродажного обслуживания продуктов на стороне заказчика // Системы высокой доступности, 2013. Т. 9. № 4. С. 25–47.
22. *Сирман М.* Стохастические модели управления запасами // Применение исследования операций в экономике. — М.: Экономика, 1977. С. 148–195.
23. *Рыжиков Ю. И.* Управление запасами. — М.: Наука, 1969. 344 с.

Поступила в редакцию 01.12.13

METHODS AND TOOLS FOR OPTIMAL PLANNING OF PROCESS PARAMETERS IN AFTERSALE SERVICE SYSTEMS OF HIGH TECHNOLOGY PRODUCTS

***I. N. Sinitsyn*[1], *A. S. Shalamov*[1], *I. V. Sergeev*[1], *E. R. Korepanov*[1], *V. V. Belousov*[1], *T. S. Gumnikova*[2], *V. S. Shorgin*[1], *and E. S. Agafonov*[1]**

[1]Institute of Informatics Problems, Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation
[2]ROSOBORONEXPORT State Corporation, 27 Stromynka Str., Moscow 107076, Russian Federation

Abstract: The methods and software tools of aftersale systems optimization are considered based on cost criteria for a given level of technical operability coefficient are considered. A brief overview of current approaches to product lifecycle management of high technology products, based on the Western concept of CALS (Continuous Acquisition and Life cycle Support) and its Russian analogue — IAS (integrated information support of products) is provided. A new stochastic approach to modeling of aftersale systems for products of high technology is presented. This approach is used to solve the actual problem of optimal design and operation of service system based on cost criteria. The methods for finding optimal values of service systems using different measures of efficiency are described. Methodological and instrumental software tools for cost optimization of spare parts supply and repairs for years to come, until

the write-off of the device under restrictions on allocated financial resources and the need to maintain a given level of technical operability coefficient, are demonstrated.

Keywords: aftersale service system; high technology product; repair and supply of spare parts; level of technical operability of manufactured product park; cost of repair and supplies; limited budget; repair and supply of spare parts program optimization; software tool

DOI: 10.14357/08696527140201

Acknowledgments

The work was financially supported by the Program of the RAS Department for Nanotechnologies and Information Technologies "Intelligent information technology, system analysis, and automation" (project 1.7).

References

1. Norenkov, I. P., and P. K. Kuz'mik. 2002. *Informatsionnaya podderzhka naukoemkikh izdeliy (CALS-tekhnologii)* [Information support of high technology products (CALS-technologies)]. Moscow: MGTU im. N. E. Baumana. 320 p.
2. Wong, K. 2008. Strategii PLM: Udlinenie ZhTsI na krupnykh servisno-orientirovannykh predpriyatiyakh [PLM strategies — longer lifecycles in service-oriented empires]. *CAD/CAM/CAE Observer* 2(38):17–20.
3. Sinitsyn, I. N., and A. S. Shalamov. 2012. *Lektsii po teorii sistem integrirovannoy logisticheskoy podderzhki* [Lectures on the theory of systems of integrated logistics support]. Moscow: TORUS PRESS. 624 p.
4. Pugachev, V. S., and I. N. Sinitsyn. 2000; 2004. *Teoriya stokhasticheskikh system* [Theory of stochastic systems]. Moscow: Logos. 1st ed. 1000 p.; 2nd ed. 1000 p.
5. Sinitsyn, I. N. 2009. *Kanonicheskie predstavleniya sluchaynykh funktsiy i ikh primenenie v zadachakh komp'yuternoy podderzhki nauchnykh issledovaniy* [Canonical expansions of random functions and their application to problems of computer support research]. Moscow: TORUS PRESS. 768 p.
6. Sinitsyn, I. N., and A. S. Shalamov. 2011. Metodologicheskie aspekty sovremennoy integrirovannoy logisticheskoy podderzhki izdeliy naukoemkoy produktsii [Methodological aspects of modern product integrated logistic support]. *Sistemy Vysokoy Dostupnosti* [Highly Available Systems] 7(4):48–74.
7. Sinitsyn, I. N., and A. S. Shalamov. 2012. Proektirovanie CALS sistem. Chast' 1. Sistemy upravleniya zhiznennym tsiklom izdeliy i ikh modelirovanie [Design of CALS Systems. Part 1. Life cycle engineering management systems and their modeling]. *Sistemy Vysokoy Dostupnosti* [Highly Available Systems] 8(3):3–17.
8. Sinitsyn, I. N., and A. S. Shalamov. 2012. Proektirovanie CALS sistem. Chast' 2. Analiticheskoe modelirovanie integrirovannykh sistem posleprodazhnogo obsluzhivaniya izdeliy naukoemkoy produktsii [Design of CALS Systems. Part 2. Analytical modeling of integrated product support systems]. *Sistemy Vysokoy Dostupnosti* [Highly Available Systems] 8(4):4–49.

9. Sinitsyn, I. N., A. S. Shalamov, and I. V. Sergeev. 2012. Problemy modelirovaniya i minimizatsii zatrat na ekspluatatsiyu izdeliy naukoemkoy produktsii na sovremennom etape [Problems of modern manufactured products modeling and operating losts minimization]. *Sbornik Dokladov XIII Mezhdunarodnoy Nauchno-Tekhnicheskoy Konferentsii "Kibernetika i Vysokie Tekhnologii XXI Veka (C&T-2012)"* [13th Science and Technology Conference (International) "Cybernetics and High Technology of the XXI Century (C&T-2012)" Proceedings]. Voronezh: Sakvoee. 2:358–370.
10. Sinitsyn, I. N., A. S. Shalamov, and V. I. Sinitsyn. 2012. Razvitie sistem integrirovannoy logisticheskoy podderzhki izdeliy naukoemkoy produktsii [Development of systems of integrated logistics support of high technology products]. *Sbornik Materialov X Mezhdunarodnoy Konferentsii "Optiko-Elektronnye Pribory i Ustroystva v Sistemakh Raspoznavaniya Obrazov, Obrabotki Izobrazheniy i Simvol'noy Informatsii (Raspoznavanie-2012)"* [10th Conference (International) "Optoelectronic Devices and Equipment in Systems for Pattern Recognition, Image, and Symbolic Information Processing (Recognition-2012)" Proceedings]. Kursk: YuZGU. 63–65.
11. Sinitsyn, I. N., A. S. Shalamov, I. V. Sergeev, V. V. Belousov, and E. S. Agafonov. 2012. Razvitie sredstv integrirovannoy logisticheskoy podderzhki izdeliy naukoemkoy produktsii na osnove sistem komp'yuternoy matematiki [Development of integrated logistics support products based on high-tech products of computer mathematics]. *Sbornik Materialov XIII Mezhdunarodnoy Nauchnoy Konferentsii "Sistemy Komp'yuternoy Matematiki i Ikh Prilozheniya"* [13th Scientific Conference (International) "Systems of Computer Mathematics and Their Applications" Proceedings]. Smolensk: SmolGU. 13:119–124.
12. Sinitsyn, I. N., A. S. Shalamov, I. V. Sergeev, V. I. Sinitsyn, E. R. Korepanov, V. V. Belousov, E. S. Agafonov, and V. S. Shorgin. 2012. Metody i sredstv analiza i modelirovaniya stokhasticheskikh sistem integrirovannoy logisticheskoy podderzhki [Methods and tools for analyzing and modeling of stochastic systems of integrated logistics support]. *Sistemy i Sredstva Informatiki — Systems and Means of Informatics* 22(2):3–28.
13. Sinitsyn, I. N., A. S. Shalamov, V. I. Sinitsyn, and E. S. Agafonov. 2013. Algoritmicheskoe i programmnoe obespechenie obrabotki informatsii i sinteza sistem integrirovannoy logisticheskoy podderzhki [Algorithms and software for information processing and integrated logistics support systems synthesis]. *Sbornik Materialov XI Mezhdunarodnoy Konferentsii "Optiko-Elektronnye Pribory i Ustroystva v Sistemakh Raspoznavaniya Obrazov, Obrabotki Izobrazheniy i Simvol'noy Informatsii (Raspoznavanie-2013)"* [11th Conference (International) "Optoelectronic Devices and Equipment in Systems for Pattern Recognition, Image, and Symbolic Information Processing (Recognition-2013)" Proceedings]. Kursk: YuZGU. 428–430.
14. Sinitsyn, I. N., and A. S. Shalamov. 2013. Proektirovanie CALS sistem. Chast' 3. Analiticheskoe modelirovanie sistem posleprodazhnogo obsluzhivaniya so smeshannymi potokami raskhodovaniya, vosstanovleniya i popolneniya zapasov [Design of CALS Systems. Part 3. Analytical modeling in contractor integrated product support systems with mixed streams of expenditure, restoring, and replenishment of inventories]. *Sistemy Vysokoy Dostupnosti* [Highly Available Systems] 9(1):4–34.
15. Sinitsyn, I. N., and A. S. Shalamov. 2013. Proektirovanie CALS sistem. Chast' 4. Statisticheskiy analiz i parametricheskiy sintez sistem posleprodazhnogo obsluzhivaniya [CALS systems design. Part 4. Statistical analysis and parametric synthesis of aftersale

product support systems]. *Sistemy Vysokoy Dostupnosti* [Highly Available Systems] 9(2):4–35.

16. Sinitsyn, I. N., A. S. Shalamov, E. R. Korepanov, V. V. Belousov, and E. S. Agafonov. 2013. Instrumental'naya sistema avtomaticheskogo poiska optimal'nykh programm postavok v sistemakh posleprodazhnogo obsluzhivaniya izdeliy [Software tools for automatic search of optimal supplement spare parts program in aftersale product support system]. *Sistemy Vysokoy Dostupnosti* [Highly Available Systems] 9(2): 47–54.
17. Sinitsyn, I. N., A. S. Shalamov, and A. A. Kuleshov. 2013. Nelineynoe korrelyatsionnoe modelirovanie i analiz nadezhnosti sistem posleprodazhnogo obsluzhivaniya izdeliy naukoemkoy produktsii [Nonlinear correlation modeling and reliability analysis of aftersale high technology product support systems]. *Sistemy i Sredstva Informatiki — Systems and Means of Informatics* 23(1):80–104.
18. Sinitsyn, I. N., A. S. Shalamov, E. R. Korepanov, V. V. Belousov, I. V. Sergeev, and A. A. Kuleshov. 2012. Razvitie algoritmicheskogo i instrumental'nogo programmnogo obespecheniya dlya analiticheskogo veroyatnostnogo modelirovaniya i optimizatsii protsessov material'no-tekhnicheskogo obespecheniya [Development of algorithmic and instrumental software tools for the stochastic analytical modeling and optimization of maintenance processes]. *Sbornik dokladov XIV Mezhdunarodnoy Nauchno-Tekhnicheskoy Konferentsii "Kibernetika i Vysokie Tekhnologii XXI Veka (C&T-2013)"* [14th Science and Technology Conference (International) "Cybernetics and High Technology of the XXI Century (C&T-2013)" Proceedings]. Voronezh: Sakvoee. 2:375–384.
19. Sinitsyn, I. N., A. S. Shalamov, V. I. Sinitsyn, E. R. Korepanov, V. V. Belousov, and A. A. Kuleshov. 2013. Metody i sredstva otsenki zapasov i urovnya gotovnosti sistem integrirovannoy logisticheskoy podderzhki, osnovannye na kanonicheskikh razlozheniyakh sluchaynykh funktsiy [Methods and tools for stock assessment and preparedness systems of integrated logistics support based on the canonical expansions of random functions]. *Materialy 3-go Mezhdunarodnogo Nauchno-Tekhnicheskogo Seminara "Sovremennye Problemy Prikladnoy Matematiki, Informatiki, Avtomatizatsii, Upravleniya"* [3rd Science and Technology Seminar (International) "Recent Developments in Applied Mathematics, Computer Science, Automation and Control" Proceedings. Moscow: IPI RAN. 115–126.
20. Sinitsyn, I. N., and A. S. Shalamov. 2013. Modelirovanie i sintez sistemy posleprodazhnogo obsluzhivaniya produktov na storone postavshchika [Modeling and design of supplier aftersale product management system]. *Sistemy Vysokoy Dostupnosti* [Highly Available Systems] 9(4):12–24.
21. Sinitsyn, I. N., and A. S. Shalamov. 2013. Modelirovanie i sintez sistemy posleprodazhnogo obsluzhivaniya produktov na storone zakazchika [Modeling and design of customer aftersale product management system]. *Sistemy Vysokoy Dostupnosti* [Highly Available Systems] 9(4):25–47.
22. Sirman, M. 1977. Stokhasticheskie modeli upravleniya zapasami [Stochastic models for inventory management]. *Primenenie issledovaniya operatsiy v ekonomike* [Application of operations research in economics]. Moscow: Ekonomika. 148–195.
23. Ryzhikov, Yu. I. 1969. *Upravlenie zapasami* [Inventory management]. Moscow: Nauka. 344 p.

Received December 01, 2013

Contributors

Sinitsyn Igor N. (b. 1940) — Doctor of Science in technology, professor, Honored scientist of RF, Head of Department, Institute of Informatics Problems, Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; sinitsin@dol.ru

Shalamov Anatoly S. (b. 1947) — Doctor of Science in technology, professor, consultant, Institute of Informatics Problems, Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; a-shal5@yandex.ru

Sergeev Igor V. (b. 1965) — Candidate of Science (PhD) in technology, Deputy Director, Institute of Informatics Problems, Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; ISergeev@ipiran.ru

Korepanov Eduard R. (b. 1966) — Candidate of Science (PhD) in technology, Head of Laboratory, Institute of Informatics Problems, Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; Ekorepanov@ipiran.ru

Belousov Vasiliy V. (b. 1977) — Candidate of Science (PhD) in technology, Head of Laboratory, Institute of Informatics Problems, Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; VBelousov@ipiran.ru

Gumnikova Tatyana S. (b. 1957) — expert, ROSOBORONEXPORT State Corporation, 27 Stromynka Str., Moscow 107076, Russian Federation; klimtat50@yandex.ru

Shorgin Vsevolod S. (b. 1978) — Candidate of Science (PhD) in technology, senior scientist, Institute of Informatics Problems, Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; VShorgin@ipiran.ru

Agafonov Egor S. (b. 1981) — scientist, Institute of Informatics Problems, Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; EAgafonov@ipiran.ru

СОГЛАСОВАНИЕ АГРЕГИРОВАННЫХ И ДЕТАЛИЗИРОВАННЫХ ПРОГНОЗОВ ПРИ РЕШЕНИИ ЗАДАЧ НЕПАРАМЕТРИЧЕСКОГО ПРОГНОЗИРОВАНИЯ*

***М. М. Стенина*[1], *В. В. Стрижов*[2]**

Аннотация: Исследуются задачи, связанные с прогнозированием большого числа временны́х рядов, образующих иерархическую структуру. К прогнозам таких рядов, как правило, предъявляется требование согласованности прогнозов по уровням иерархии. В статье предлагается алгоритм согласования прогнозов иерархических временных рядов, основанный на решении задачи оптимизации с ограничениями. Предлагаемый алгоритм позволяет проводить согласование прогнозов в случае неплоской иерархической структуры, а также учитывать физические ограничения на прогнозируемые величины, такие как неотрицательность или максимальное значение. Работа алгоритма демонстрируется на данных посуточной загруженности железнодорожных узлов в Омской области, качество прогнозов сравнивается с качеством прогнозов алгоритма оптимального согласования. Также демонстрируется работа предлагаемого алгоритма при неплоской иерархической структуре временны́х рядов.

Ключевые слова: иерархические временны́е ряды; непараметрическое прогнозирование; эмпирическое распределение; согласование прогнозов

DOI: 10.14357/08696527140202

1 Введение

Во многих прикладных областях часто возникают задачи, связанные с прогнозированием большого числа временны́х рядов, образующих иерархическую структуру, основанную, например, на разбиении на географические районы или по группам товаров или грузов. Основное требование, предъявляемое к прогнозам иерархических временны́х рядов, — согласованность сумм прогнозируемых величин по уровням иерархии.

Существует несколько основных подходов к задаче согласования прогнозов иерархических временны́х рядов. Подходы top-down и bottom-up являются самыми распространенными. Подход top-down предполагает получение прогноза на верхнем уровне иерархии (агрегированный временной ряд), а затем

*Работа выполнена при финансовой поддержке РФФИ (проект № 13-07-13139).

[1]Московский физико-технический институт, mmedvednikova@gmail.com

[2]Вычислительный центр им. А. А. Дородницына Российской академии наук, strijov@gmail.com

деагрегирование этого прогноза на следующий (более низкий) уровень иерархии на основании долей, наблюдаемых в истории. Подход bottom-up использует прогнозы временны́х рядов нижнего уровня иерархии (неагрегированных), из которых получает прогнозы рядов из верхних уровней путем агрегирования. Также встречаются подходы, комбинирующие top-down и bottom-up, их называют middle-out.

Нет единой точки зрения на то, какой из подходов позволяет получать более точные прогнозы. Наиболее ранние исследования проведены в работе [1], где авторы утверждают, что неагрегированные данные содержат много ошибок и поэтому top-down прогнозирование дает более точные прогнозы. К таким же выводам приходят авторы работ [2, 3]. В [4] также доказывается, что агрегированные прогнозы более точны. С другой стороны, в [5, 6] показано, что основные потери информации происходят при агрегировании и поэтому bottom-up подход предпочтительнее. В [7] сравниваются оба подхода к согласованию прогнозов и утверждается, что bottom-up предпочтительнее при выполнении некоторых условий на структуру иерархии и горизонт прогноза. В [8] исследуются смещение и устойчивость прогнозов, получаемых с помощью обоих подходов, и заключается, что bottom-up надежнее, за исключением случаев с пропусками значений и выбросами на нижних уровнях иерархии.

В следующих работах проводились теоретические исследования подходов bottom-up и top-down, которые не дали выводов о том, что один из подходов согласования прогнозов более предпочтителен, чем другой. В [9, 10] приводятся теоретические аргументы в пользу того, что точность прогнозов зависит от ковариационной структуры рядов — компонентов иерархической структуры. В работе [11] обсуждается несколько моделей временны́х рядов и демонстрируется, что нет однозначного превосходства одного подхода над другим. В [12] заключается, что необходимо комбинировать оба подхода. В [13] обобщаются направления прогнозирования иерархических временны́х рядов, но не предлагается новых подходов.

Авторы статьи [14] обобщают существующие подходы к согласованию иерархических прогнозов и предлагают оптимальное согласование с использованием регрессии. В этой работе утверждается, что предлагаемый подход охватывает все возможные способы согласования прогнозов для иерархических временны́х рядов. Однако в статье рассматриваются только способы согласования, включающие всевозможные суммирования прогнозов рядов нижнего уровня иерархии для получения прогнозов верхних уровней либо разбиения прогнозов верхнего уровня иерархии для получения прогнозов нижнего уровня.

В настоящей работе предлагается принципиально иной подход к задаче согласования прогнозов иерархических временны́х рядов, позволяющий проводить согласование не только для иерархий с плоской структурой, но и для иерархий, в которых разбиение производится более чем по одной размерности (например, разбиение по товарам и территориальное разбиение по местоположению магазинов при прогнозировании спроса или разбиение по типам грузов и же-

лезнодорожным веткам при прогнозировании объемов грузоперевозок). Кроме того, при использовании предлагаемого подхода есть возможность учитывать физические ограничения на прогнозируемые величины, такие как, например, их неотрицательность. Корректировка прогнозов производится путем оптимизации квадратичного функционала с ограничениями типа равенства и неравенства. Ограничение-равенство соответствует требованию равенства прогноза верхнего уровня иерархии сумме прогнозов нижнего уровня иерархии, а ограничения-неравенства связаны с физической природой прогнозируемых величин (например, количество проданного товара или отправленных вагонов не может быть отрицательным). При многоуровневой иерархической структуре согласование проводится сверху вниз, т. е. сперва согласуются первый и второй уровни, затем второй и третий и т. д.

Для оценки качества работы предлагаемой процедуры согласования проводится эксперимент по прогнозированию объемов железнодорожных грузоперевозок 38 типов грузов по 99 веткам. Для варианта плоской иерархической структуры проводится сравнение качества прогноза с алгоритмом оптимального согласования [14], который, как показано в [14], превосходит по качеству прогнозов подходы top-down и bottom-up. Также оценивается качество прогноза при согласовании для иерархической структуры с двумя размерностями (ветки и типы грузов).

В качестве алгоритма прогнозирования временны́х рядов выбран алгоритм непараметрического прогнозирования Hist, описанный в [15].

В разд. 2 рассматривается предлагаемый алгоритм согласования, затем в разд. 3 описывается используемый алгоритм прогнозирования, разд. 4 посвящен оценке качества работы предлагаемого алгоритма и его сравнению с алгоритмом оптимального согласования [14].

2 Задача согласования прогнозов

2.1 Согласование прогнозов при плоской иерахической структуре

Рассмотрим двухуровненую плоскую иерархическую структуру, в которой на верхнем уровне находится один временной ряд

$$\mathbf{X} = \{X_t\}, \quad t = 1, \ldots, T.$$

Пусть на нижнем уровне находится n временны́х рядов

$$\mathbf{x}_i = \{x_{it}\}, \quad t = 1, \ldots, T, \quad i = 1, \ldots, n.$$

Связь между рядом верхнего уровня и рядами нижнего уровня задается соотношением

$$X_t = \sum_{i=1}^{n} x_{it}, \quad t = 1, \ldots, T.$$

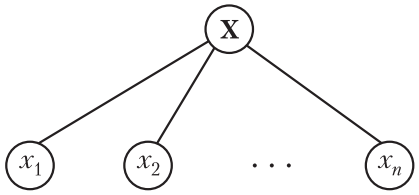


Рис. 1 Двухуровневая плоская иерархическая структура

Графически эта структура изображена на рис. 1.

Пусть известны прогнозы для временны́х рядов нижнего уровня иерархии $\hat{x}_1, \ldots, \hat{x}_n$ и для каждого прогноза сделана оценка доверительного интервала $[\hat{x}_1 - d_1;\, \hat{x}_1 + d_1], \ldots, [\hat{x}_n - d_n;\, \hat{x}_n + d_n]$.

Пусть прогноз для временно́го ряда верхнего уровня иерархии равен $\hat{X}$. Предлагается, не корректируя прогноз верхнего уровня $\hat{X}$, построить скорректированные прогнозы y_i, которые являются решением оптимизационной задачи с ограничениями

$$\left.\begin{aligned} Q = \sum_{i=1}^{n} \frac{1}{d_i^2} (y_i - \hat{x}_i)^2 \longrightarrow \min\,; \\ \sum_{i=1}^{n} y_i = \hat{X}\,; \quad y_i \geq 0\,, \quad i = 1, \ldots, n\,. \end{aligned}\right\} \tag{1}$$

Скорректированные прогнозы y_i должны быть максимально близки к исходным прогнозам $\hat{x}_i$ в смысле квадратичного отклонения Q и при суммировании давать прогноз $\hat{X}$, т. е. быть с ним согласованными. Ограничения-неравенства в этой оптимизационной задаче соответствуют физическим ограничениям на прогнозируемую величину (в данном случае рассматриваются только неотрицательные величины).

При отсутствии активных ограничений-неравенств задача (1) имеет аналитическое решение

$$y_i = \hat{x}_i + \frac{d_i^2}{\sum\limits_{i=1}^{n} d_i^2} \left(\hat{X} - \sum_{i=1}^{n} \hat{x}_i \right) ,$$

которое имеет явный физический смысл: рассогласованность прогнозов $\hat{X} - \sum\limits_{i=1}^{n} \hat{x}_i$ распределяется по компонентам пропорционально ширине их доверительных интервалов. При активных ограничениях-неравенствах необходимо решать задачу (1) с помощью итерационного процесса. Этот процесс сходится достаточно быстро в силу выпуклости функционала Q и, следовательно, наличия у него единственного глобального минимума.

При наличии в иерархической структуре более двух уровней предлагается проводить описанную процедуру согласования по уровням от наименее детализированных к более детализированным (сверху вниз). При этом согласование в

каждом узле дерева, соответсвующего иерархической структуре, рассматривается как отдельная оптимизационная задача (1).

2.2 Согласование прогнозов при иерархической структуре с разбиением более чем по одному измерению

Рассмотрим иерархическую структуру, в которой разбиение проводится по двум размерностям. Например, для объемов железнодорожных перевозок такие разбиения можно проводить по веткам и типам грузов. В первом уровне этой структуры находится временной ряд $\mathbf{S}$ (рис. 2), соответствующий суммарному отправлению всех типов грузов со всех веток. Обозначим число веток через m, а количество типов грузов через n. Если разбиение общего количества вагонов производится по веткам, то каждый ряд второго уровня $\mathbf{X}_j$, $j = 1, \ldots, m$, соответствует отправлению с ветки j всех типов грузов. Если разбиение проводится по грузам, то каждый ряд второго уровня $\mathbf{Z}_i$, $i = 1, \ldots, n$, соответствует отправлению вагонов с типом груза i со всех веток. На третьем уровне иерархии разбиение проводится одновременно по веткам и типам грузов, поэтому у временны́х рядов $\mathbf{x}_{ij}$ двойная индексация. Каждый ряд $\mathbf{x}_{ij}$ соответствует отправлению груза i с ветки j. На рис. 2 эти ряды разнесены для различных разбиений на втором уровне для наглядности, но в действительности это один и тот же набор временны́х рядов.

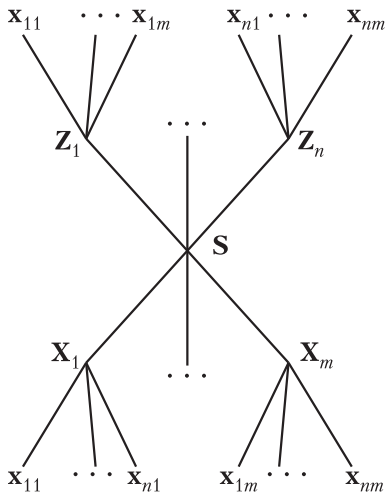


Рис. 2 Иерархическая структура с разбиением по двум измерениям

Соотношения, связывающие значения временны́х рядов на разных уровнях иерархии, задаются следующим образом:

$$Z_{it} = \sum_{j=1}^{m} x_{ijt}\,, \quad t = 1, \ldots, T\,, \quad i = 1, \ldots, n\,;$$

$$X_{jt} = \sum_{i=1}^{n} x_{ijt}\,, \quad t = 1, \ldots, T\,, \quad j = 1, \ldots, m\,;$$

$$S_t = \sum_{i=1}^{n} Z_{it} = \sum_{j=1}^{m} X_{jt}\,, \quad t = 1, \ldots, T\,.$$

Пусть известны прогнозы $\hat{S},\ \hat{X}_1,\dots,\hat{X}_m,\ \hat{Z}_1,\dots,\hat{Z}_n,\ \hat{x}_{11},\ \hat{x}_{12},\dots,\hat{x}_{nm}$ и доверительные интервалы $[\hat{X}_1 - D_1; \hat{X}_1 + D_1],\ \dots,\ [\hat{X}_m - D_m;\ \hat{X}_m + D_m]$, $[\hat{Z}_1 - E_1; \hat{Z}_1 + E_1],\ \dots,\ [\hat{Z}_n - E_n; \hat{Z}_n + E_n],\ [\hat{x}_{11} - d_{11}; \hat{x}_{11} + d_{11}]$, $[\hat{x}_{12} - d_{12}; \hat{x}_{12} + d_{12}],\dots,[\hat{x}_{nm} - d_{nm};\ \hat{x}_{nm} + d_{nm}]$.

Требуется построить скорректированные прогнозы y_{ij}, которые являются решением оптимизационной задачи с ограничениями

$$\left.\begin{aligned} Q = \sum_{i=1}^{n}\sum_{j=1}^{m}\frac{1}{d_{ij}^2}\,(y_{ij} - \hat{x}_{ij})^2 \longrightarrow \min\,; \\ \sum_{i=1}^{n} y_{ij} = \hat{X}_j\,, \quad j = 1,\dots,m\,; \\ \sum_{j=1}^{m} y_{ij} = \hat{Z}_i\,, \quad i = 1,\dots,n\,; \\ y_{ij} \geq 0\,, \quad i = 1,\dots,n\,, \quad j = 1,\dots,m\,. \end{aligned}\right\} \tag{2}$$

Необходимым условием существования решения задачи (2) является согласованность прогнозов второго уровня $\hat{X}_j$, $\hat{Z}_i$:

$$\sum_{i=1}^{n}\hat{Z}_i = \sum_{j=1}^{m}\hat{X}_j\,. \tag{3}$$

Поскольку это условие для прогнозов, полученных независимо для разных временны́х рядов не будет выполняться почти никогда, необходимо найти скорректированные прогнозы Y_j, W_i, т. е. решить оптимизационную задачу с ограничениями

$$\left.\begin{aligned} Q = \sum_{j=1}^{m}\frac{1}{D_j^2}\left(Y_j - \hat{X}_j\right)^2 + \sum_{i=1}^{n}\frac{1}{E_i^2}\left(W_i - \hat{Z}_i\right)^2 \longrightarrow \min\,; \\ \sum_{j=1}^{m} Y_j = \sum_{i=1}^{n} W_i; \\ Y_j \geq 0\,, \quad j = 1,\dots,m\,; \\ W_i \geq 0\,, \quad i = 1,\dots,n\,. \end{aligned}\right\} \tag{4}$$

Затем найденные скорректированные прогнозы Y_j, W_i необходимо подставить в (2) вместо прогнозов $\hat{X}_j$, $\hat{Z}_i$ и найти скорректированные прогнозы y_{ij}.

При отсутствии активных ограничений-неравенств аналитическое решение задачи (2) записывается аналогично решению задачи (1), в противном случае решение находится с помощью итерационного процесса.

Для иерархических структур с разбиением по большему числу измерений оптимизационная задача согласования прогнозов записывается аналогично задаче (2) и требует для выполнения условия (3) решения оптимизационных задач, подобных задаче (4).

3 Алгоритм непараметрического прогнозирования Hist

Рассмотрим задачу согласования прогнозов, полученных с помощью алгоритма Hist. Этот алгоритм подробно рассматривается в статье [15]. Приведем здесь краткое описание варианта этого алгоритма, использованного в настоящей статье.

По заданному временно́му ряду $\mathbf{x}$ строится гистограмма $\mathcal{H}$ — набор пар

$$\mathcal{H} = \{(y_k, g_k)\}_{k=1}^{K},$$

где K — число интервалов $y_k = [y_k^{\min}, y_k^{\max}]$ со средним значением $\bar{y}_k$, на которые разбита ось значений ряда; g_k — высота столбца гистограммы на интервале y_k, которая равна количеству точек ряда, попавших в этот интервал. Предполагается, что рассматриваемый временной ряд $\mathbf{x}$ стационарен.

Выберем границы гистограммы, число столбцов и разбиение на столбцы следующим образом:

(1) пусть n — число точек x_t с известными значениями временно́го ряда;

(2) выберем число столбцов (обоснование см. в [16]) $K = \lceil 3\sqrt[3]{n}\,\rceil$, если $K < 5$, то $K = 5$, если $K > 100$, то $K = 100$;

(3) границы $y_1^{\min} = \min\limits_{t=1,\ldots,T}(x_t)$, $y_k^{\max} = \max\limits_{t=1,\ldots,T}(x_t)$;

(4) столбцы выбираются равной ширины.

Для каждого $k = 1, \ldots, K$ высота столбца гистограммы g_k равна

$$g_k = \sum_{t=1}^{T} [x_t \in y_k] \ ,$$

где индикаторная функция $[\cdot]$ равна 1, если в скобках стоит истинное логическое выражение, и 0 в противном случае.

Алгоритм непараметрического прогнозирования Hist

Введем функцию потерь $L(\hat{x}, x)$ — штраф за несоответствие прогнозируемого значения $\hat{x}$ историческому значению x.

Прогнозируемое значение ряда $\hat{x}$ находится как значение $\hat{x} \in \{\bar{y}_1, \ldots, \bar{y}_K\}$, соответствующее оптимальному значению свертки распределения $\{g_k\}_{k=1}^K$ и функции потерь L:

$$\hat{x} = \underset{z \in \{\bar{y}_1, \ldots, \bar{y}_K\}}{\operatorname{argmin}} \sum_{k=1}^{K} g_k L(z, \bar{y}_k)\,.$$

В текущей работе использовалась абсолютная функция потерь

$$L(\hat{x}, x) = |\hat{x} - x|\ .$$

Рассмотрим особенности задачи согласования прогнозов, связанные с выбором алгоритма прогнозирования Hist. Доверительные интервалы полученных прогнозов естественным образом определяются шириной столбцов используемых гистограмм, а именно:

$$[\hat{x} - d;\, \hat{x} + d] = \left[y_k^{\min};\, y_k^{\max}\right]\ ,$$

поэтому

$$d = 0{,}5(y_k^{\max} - y_k^{\min}).$$

4 Вычислительный эксперимент

4.1 Экспериментальные данные

В эксперименте использованы данные о посуточной загруженности железнодорожных узлов РЖД с 1 января 2007 г. по 22 апреля 2008 г. В табл. 1 приведен пример записи базы данных.

Коды станций представляют собой шестизначные числа. Станции, в коде которых две первые цифры совпадают, входят в одну железнодорожную ветку. Станций отправления 1566, станций назначения 1902, веток 99. Код груза — натуральное число от 1 до 38; также имеются перевозки, где код груза не указан. Род вагона — натуральное число, в имеющихся данных 75 различных родов вагонов.

Для прогноза были использованы временны́е ряды отправления вагонов с различными типами грузов по каждой ветке. Временны́е ряды сильно варьируются по средним и максимальным значениям и дисперсии значений. Также

Таблица 1 Вид записи базы данных железнодорожных перевозок

Дата погрузки	Станция отправления	Станция назначения	Количество вагонов	Код груза	Род вагона	Суммарный вес груза	Признак маршрутной отправки
2007-01-01	020108	932902	1	1	216	56	9

существенная часть рядов — постоянные ряды, все элементы которых равны нулю.

4.2 Оценка качества прогноза

Поскольку прогнозируемые временны́е ряды значительно отличаются друг от друга, необходимо использовать способ оценки качества прогноза, позволяющий проводить сравнение качества для разнородных рядов. Поэтому оценивалось отношение абсолютной ошибки прогноза (MAE — mean absolute error) к максимально возможной ошибке. При использовании алгоритма прогнозирования Hist максимально возможная ошибка прогноза — это разница между максимальным и минимальным значениями временно́го ряда:

$$\text{error} = \frac{|\hat{x} - x|}{\max\limits_{t=1,\ldots,T} x_t - \min\limits_{t=1,\ldots,T} x_t} = \frac{\text{MAE}}{\max\limits_{t=1,\ldots,T} x_t - \min\limits_{t=1,\ldots,T} x_t}. \qquad (5)$$

Если разница между максимальным и минимальным значениями ряда равна нулю, то предполагалось, что ошибка прогноза такого ряда равна нулю, так как метод Hist прогнозирует постоянный временной ряд его значением. Ошибка усреднялась по 100 контрольным точкам (100 последних из 478 точек истории).

Оценивалось качество прогноза до и после процедуры согласования прогнозов.

4.3 Согласование прогнозов для плоской иерархической структуры

Рассматривались группы временны́х рядов, образующих иерархические структуры такого вида, как на рис. 1, с двумя уровнями иерархии. Поскольку структура экспериментальных данных позволяет разбивать временны́е ряды как по типам грузов, так и по веткам, были рассмотрены следующие задачи согласования прогнозов.

1. Согласование прогноза суммарного количества отправленных с ветки вагонов со всеми типами грузов и прогнозов отправления всех типов грузов по отдельности с заданной ветки (всего 99 задач согласования прогнозов, в каждой на верхнем уровне иерархии один временной ряд, на нижнем $n = 38$ рядов).
2. Согласование прогноза общего количества вагонов с заданным грузом, отправленных со всех веток, с прогнозом отправления вагонов с этим же грузом со всех веток по отдельности (всего 38 задач согласования прогнозов, в каждой на верхнем уровне иерархии один временной ряд, на нижнем $n = 99$ временны́х рядов).

Средние ошибки прогноза (формула (5)), полученные для независимых прогнозов, для алгоритма оптимального согласования из [14] и для предлагаемого

Таблица 2 Средние ошибки прогнозов для согласования по грузам

Уровень иерархии	Независимые прогнозы	Оптимальное согласование [14]	Предлагаемый алгоритм
Верхний	**0,0502**	0,0579	**0,0502**
Нижний	0,0059	0,0081	**0,0058**

Таблица 3 Средние ошибки прогнозов для согласования по веткам

Уровень иерархии	Независимые прогнозы	Оптимальное согласование [14]	Предлагаемый алгоритм
Верхний	0,0846	**0,0579**	0,0846
Нижний	0,0059	0,0062	**0,0058**

алгоритма представлены в табл. 2 и 3. Поскольку алгоритм [14] не позволяет учитывать то, что прогнозируемые величины неотрицательны, полученные при согласовании отрицательные прогнозы обнулялись. При решении задачи оптимизации (1) прогнозы постоянных временны́х рядов (для которых максимально возможная ошибка прогноза равна нулю) исключались из рассмотрения и не корректировались.

В работе [14] было отмечено, что на практике алгоритм оптимального согласования позволяет уменьшить ошибку прогнозирования рядов из верхних уровней иерархии, но увеличивает ошибку прогноза временны́х рядов из нижнего уровня иерархии по сравнению с независимыми прогнозам всех временны́х рядов. Проведенный эксперимент показывает, что для используемых экспериментальных данных эта тенденция сохраняется, в то время как согласование прогнозов с помощью решения оптимизационной задачи (1) позволяет проводить согласование без увеличения ошибки прогнозирования на обоих уровнях иерархии.

4.4 Согласование прогнозов для неплоской иерархической структуры

Предлагаемый алгоритм, в отличие от алгоритма оптимального согласования [14], позволяет проводить согласование прогнозов для неплоских иерархических структур. Рассматривалась иерархическая структура, изображенная на рис. 2. Временной ряд S в корне структуры соответствует отправлению вагонов со всеми типами грузов со всех веток. На следующем уровне этот ряд разделялся:

(1) по грузам ($m = 38$ временны́х рядов, соответствующих отправлению каждого из 38 грузов со всех веток);

(2) по веткам ($n = 99$ временны́х рядов, соответствующих отправлению с каждой из 99 веток всех грузов).

На третьем уровне иерархии разбиение происходит одновременно по типам грузов и веткам (38×99 временны́х рядов).

Таблица 4 Средние ошибки прогнозов при неплоской иерархической структуре

Уровень	Независимые прогнозы	Предлагаемый алгоритм
Верхний уровень по веткам	0,0502	0,0503
Верхний уровень по грузам	0,0846	0,0845
Нижний уровень	0,0059	0,0150

Для согласования прогнозов были решены оптимизацинные задачи (2) и (4). Ошибка прогноза оценивалась также по формуле (5) на втором и третьем уровнях иерархии. Результаты представлены в табл. 4, из которой видно, что ошибка прогноза на верхнем уровне иерархии практически не изменяется после согласования прогнозов, а на нижнем уровне иерархии есть ухудшение качества прогнозов. Таким образом, поведение предлагаемого алгоритма схоже с поведением алгоритма из [14], который не применим для иерархической структуры такого типа.

5 Заключение

Предложен алгоритм согласования прогнозов иерархических временны́х рядов, основанный на решении оптимизационной задачи с ограничениями. Предлагаемый алгоритм позволяет проводить согласование прогнозов в случае сложной неплоской иерархической структуры временны́х рядов, а также позволяет в ходе процедуры согласования учитывать физические ограничения на прогнозируемые величины. Проведенный эксперимент показал, что предлагаемый алгоритм не уступает по качеству прогнозов алгоритму оптимального согласования и при этом охватывает более широкий класс задач, нежели алгоритм оптимального согласования.

Литература

1. *Grunfeld Y., Griliches Z.* Is aggregation necessarily bad? // Rev. Econ. Stat., 1960. Vol. 42. No. 1. P. 1–13.
2. *Fogarty D. W., Blackstone J. H., Hoffman T. R.* Production and inventory management. — 2nd ed. — Cincinnati, OH: South-Western Publication Co., 1990. 880 p.
3. *Narasimhan S. L., McLeavey D. W., Billington P. J.* Production planning and inventory control. — 2nd ed. — Englewood Cliffs, NJ: Prentice Hall, 1995. 716 p.
4. *Fliedner G.* An investigation of aggregate variable time series forecast strategies with specific subaggregate time series statistical correlation // Comput. Oper. Res., 1999. Vol. 26. No. 10–11. P. 1133–1149.
5. *Orcutt G. H., Watts H. W., Edwards J. B.* Data aggregation and information loss // Amer. Econ. Rev., 1968. Vol. 58. No. 4. P. 773–787.
6. *Edwards J. B., Orcutt G. H.* Should aggregation prior to estimation be the rule? // Rev. Econ. Stat., 1969. Vol. 51. No. 4. P. 409–420.

7. *Shlifer E., Wolff R. W.* Aggregation and proration in forecasting // Management Sci., 1979. Vol. 25. No. 6. P. 594–603.
8. *Schwarzkopf A. B., Tersine R. J., Morris J. S.* Top-down versus bottom-up forecasting strategies // Int. J. Prod. Res., 1998. Vol. 26. No. 11. P. 1833–1843.
9. *Tiao G. C., Guttman I.* Forecasting contemporal aggregates of multiple time series // J. Econometrics, 1980. Vol. 12. No. 2. P. 219–230.
10. *Kohn R.* When is an aggregate of a time series effciently forecast by its past? // J. Econometrics, 1982. Vol. 18. No. 3. P. 337–349.
11. *Shing N. K.* A study of bottom-up and top-down forecasting methods. M.Sc. Thesis. — Melbourne: Royal Melbourne Institute of Technology, 1993.
12. *Kahn K. B.* Revisiting top-down versus bottom-up forecasting // J. Business Forecasting, 1998. Vol. 17. No. 2. P. 14–19.
13. *Fliedner G.* Hierarchical forecasting: Issues and use guidelines // Ind. Management Data Syst., 2001. Vol. 101. No. 1. P. 5–12.
14. *Hyndman R. J., Ahmed R. A., Athanasopoulos G., Shang H. L.* Optimal combination forecasts for hierarchical time series // Comp. Stat. Data Anal., 2011. Vol. 55. No. 9. P. 2579–2589.
15. *Вальков А. С., Кожанов Е. М., Медведникова М. М., Хусаинов Ф. И.* Непараметрическое прогнозирование загруженности системы железнодорожных узлов по историческим данным // Машинное обучение и анализ данных, 2012. Т. 1. Вып. 4. С. 448–465.
16. *Scott D. W.* On optimal and data-based histograms // Biometrika, 1979. Vol. 66. No. 3. P. 605–610.

Поступила в редакцию 31.03.14

RECONCILIATION OF AGGREGATED AND DISAGGREGATED TIME SERIES FORECASTS IN NONPARAMETRIC FORECASTING PROBLEMS

***M. M. Stenina*[1] *and V. V. Strijov*[2]**

[1]Moscow Institute of Physics and Technology, 9 Institutskiy Per., Dolgoprudny, Moscow Region 141700, Russian Federation
[2]Dorodnicyn Computing Center, Russian Academy of Sciences, 40 Vavilov Str., Moscow 119333, Russian Federation

Abstract: In many applications, there are the problems of forecasting a lot of time series with hierarchical structure. It is needed to reconcile forecasts across the hierarchy. In this paper, a new algorithm of reconciliation of hierarchical time series forecasts is proposed. This algorithm is based on solving the optimization problem with constraints. The proposed algorithm allows to reconcile the forecasts with nonplanar hierarchical structure and to take into account physical constraints

of forecasted values such as nonnegativeness or maximal value. The algorithm performance is illustrated by the railroad stations occupancy data in the Omsk region. The quality of forecasts is compared with the quality of forecasts made by the optimal algorithm of reconciliation. Also, the algorithm performance is demonstrated for the nonplanar hierarchical structure of time series.

Keywords: hierarchical time series; nonparametric forecasting; empirical distribution; forecasts reconciliation

DOI: 10.14357/08696527140202

Acknowledgments

The work was financially supported by the Russian Foundation for Basic Research (project No. 13-07-13139).

References

1. Grunfeld, Y., and Z. Griliches. 1960. Is aggregation necessarily bad? *Rev. Econ. Stat.* 42(1):1–13.
2. Fogarty, D. W., J. H. Blackstone, and T. R. Hoffman. 1990. *Production and inventory management.* 2nd ed. Cincinnati, OH: South-Western Publication Co. 880 p.
3. Narasimhan, S. L., D. W. McLeavey, and P. J. Billington. 1995. *Production planning and inventory control*. 2nd ed. Englewood Cliffs, NJ: Prentice Hall. 716 p.
4. Fliedner, G. 1999. An investigation of aggregate variable time series forecast strategies with specific subaggregate time series statistical correlation. *Comput. Oper. Res.* 26(10-11):1133–1149.
5. Orcutt, G. H., H. W. Watts, and J. B. Edwards. 1968. Data aggregation and information loss. *Amer. Econ. Rev.* 58(4):773–787.
6. Edwards, J. B., and G. H. Orcutt. 1969. Should aggregation prior to estimation be the rule? *Rev. Econ. Stat.* 51(4):409–420.
7. Shlifer, E., and R. W. Wolff. 1979. Aggregation and proration in forecasting. *Management Sci.* 25(6):594–603.
8. Schwarzkopf, A. B., R. J. Tersine, and J. S. Morris. 1988. Top-down versus bottom-up forecasting strategies. *Int. J. Prod. Res.* 26(11):1833–1843.
9. Tiao, G. C., and I. Guttman. 1980. Forecasting contemporal aggregates of multiple time series. *J. Econometrics* 12(2):219–230.
10. Kohn, R. 1982. When is an aggregate of a time series effciently forecast by its past? *J. Econometrics* 18(3):337–349.
11. Shing, N. K. 1993. A study of bottom-up and top-down forecasting methods. M.Sc. Thesis. Melbourne: Royal Melbourne Institute of Technology.
12. Kahn, K. B. 1998. Revisiting top-down versus bottom-up forecasting. *J. Business Forecasting* 17(2):14–19.
13. Fliedner, G. 2001. Hierarchical forecasting: Issues and use guidelines. *Ind. Management Data Syst*. 101(1):5–12.

14. Hyndman, R. J., R. A. Ahmed, G. Athanasopoulos, and H. L. Shang. 2011. Optimal combination forecasts for hierarchical time series. *Comput. Stat. Data Anal.* 55(9):2579–2589.
15. Val'kov, A. S., E. M. Kozhanov, M. M. Medvednikova, and F. I. Khusainov. 2012. Neparametricheskoe prognozirovanie zagruzhennosti sistemy zheleznodorozhnykh uzlov po istoricheskim dannym [Nonparametric forecasting of railroad stations occupancy according to historical data]. *Mashinnoe Obuchenie i Analiz Dannykh* [J. Machine Learning and Data Analysis] 1(4):448–465.
16. Scott D. W. 1979. On optimal and data-based histograms. *Biometrika* 66(3):605–610.

Received March 31, 2014

Contributors

Stenina Mariya M. (b. 1991) — student, Moscow Institute of Physics and Technology, 9 Institutskiy per., Dolgoprudny, Moscow Region 141700, Russian Federation; mmedvednikova@gmail.com

Strijov Vadim V. (b. 1967) — Candidate of science (PhD) in physics and mathematics, associate professor, scientist, Dorodnicyn Computing Center, Russian Academy of Sciences, 40 Vavilov Str., Moscow 119333, Russian Federation; strijov@gmail.com

ОЦЕНИВАНИЕ ЭФФЕКТИВНОЙ ПРОПУСКНОЙ СПОСОБНОСТИ УЗЛА В ИНФОКОММУНИКАЦИОННОЙ ТАНДЕМНОЙ СЕТИ*

А. В. Бородина[1], *Е. В. Морозов*[2]

Аннотация: Исследуются свойства регенеративной оценки эффективной пропускной способности (ЭПС) коммуникационного узла тандемной сети. Ранее этот вопрос был рассмотрен для отдельного узла с входным регенерирующим процессом. Такая постановка задачи является естественной для ациклических сетей, так как входной процесс восстановления, проходя через узлы такой сети в условиях стационарности, оказывается положительно возвратным регенерирующим процессом. На основе результатов теории больших уклонений в работе предложена аппроксимация величины ЭПС, качество которой проверено моделированием на примере ряда сетей тандемного типа. Для времени обслуживания и величины нагрузки использовались распределение Вейбулла с легким хвостом, усеченное распределение Парето, а также экспоненциальное распределение, а число узлов в сети варьировалось от 2 до 40. Показано, что полученная на основе аппроксимации оценка ЭПС обеспечивает выполнение условия: оценка вероятности превышения процессом нагрузки заданного уровня всегда меньше заданного значения этой вероятности (гарантии качества сервиса, QoS — quality of service). Этот результат указывает на возможность использования предложенной аппроксимации при выборе величины ЭПС узлов инфокоммуникационных тандемных сетей высокой надежности.

Ключевые слова: тандемная сеть; эффективная пропускная способность; регенерирующий входной процесс; качество обслуживания; теория больших уклонений; аппроксимация; статистическое оценивание; имитационное моделирование

DOI: 10.14357/08696527140203

1 Введение

В классической теории систем обслуживания мощность C обслуживающего прибора, т. е. величина работы, которую прибор может выполнить за единицу

*Работа выполнена при финансовой поддержке Программы стратегического развития ПетрГУ в рамках реализации комплекса мероприятий по развитию научно-исследовательской деятельности.

[1]Институт прикладных математических исследований Карельского научного центра Российской академии наук; Петрозаводский государственный университет, borodina@krc.karelia.ru

[2]Институт прикладных математических исследований Карельского научного центра Российской академии наук; Петрозаводский государственный университет, emorozov@karelia.ru

времени, как правило, считается заданной и равной 1. Однако для реальных вычислительных и инфокоммуникационных систем значение мощности C необходимо выбирать, чтобы обеспечить требуемый регламентированный уровень обслуживания, обобщенно называемый качеством обслуживания (QoS) (например, время задержки, величина джиттера, вероятность потери данных и т. п.).

Рассмотрим систему с одним сервером, постоянной скоростью обслуживания C и конечным размером буфера b, на вход которой поступает внешний поток нагрузки. Пусть W обозначает стационарную нагрузку, т. е. *типичную незавершенную работу* при функционировании системы в устойчивом режиме. Если требованием QoS является условие

$$\mathsf{P}_b := \mathsf{P}(W > b) \le \Gamma\,, \tag{1}$$

где Γ — заданная величина (гарантия QoS), то обеспечивающая это требование скорость C называется *эффективной пропускной способностью* сервера. (Варианты этого понятия, связанные с применениями в различных коммуникационных системах, подробно обсуждаются в [1].)

Пусть v_i есть величина нагрузки, поступающей в систему в интервале $[i, i+1)$. Предположим, что в некоторой положительной окрестности $(0, \theta_0)$ параметра θ существует конечный (при всех $\theta \in (0,\, \theta_0)$) предел

$$\Lambda_V^{(n)}(\theta) := \frac{1}{n} \ln \mathsf{E} e^{\theta \sum\limits_{i=0}^{n-1} v_i} \to \Lambda_V(\theta)\,,\ \ n \to \infty\,. \tag{2}$$

Этот предел называется (нормированной) предельной логарифмической производящей функцией моментов входного процесса [1, 2]. В общем случае предполагается, что последовательность $\{v_i\}$ является стационарной с перемешиванием, а для существования предела требуется выполнение условия

$$\mathsf{E} e^{\theta v} < \infty\,,$$

где v — типичный элемент последовательности $\{v_i\}$ (т. е. случайная величина (с.в.) v распределена как любая с.в. v_i) [2, 3]. Тогда ЭПС определяется как (см., например, [2])

$$C := \frac{\Lambda_V(\theta^*)}{\theta^*}\,, \tag{3}$$

где искомый параметр θ^* может быть найден из условия (1) и принципа больших уклонений для стационарного процесса W, который имеет форму:

$$\lim_{x\to\infty} \frac{1}{x} \ln \mathsf{P}(W > x) = -\theta^*.$$

(Точные условия выполнения этого принципа для процессов обслуживания можно найти, например, в [4, 5].) Вместе с (1) это дает такое значение искомого параметра θ^*,

$$\theta^* = -\frac{\ln \Gamma}{b} > 0\,,$$

подстановка которого в (3) позволяет найти требуемое значение ЭПС.

В предшествующих работах [6–8] была исследована регенеративная оценка ЭПС в случае отдельного узла, на вход которого поступает регенерирующая последовательность $\{v_i\}$. В данной работе этот анализ распространен на коммуникационные узлы, образующие тандемную сеть.

2 Методы оценивания эффективной пропускной способности

Вычислить предел (2) аналитически можно лишь в простейших случаях. (В этой связи укажем работы [1, 7, 8].) Например, если входная последовательность $\{v_i\}$ состоит из независимых одинаково распределенных (н.о.р.) с.в., то $\Lambda_V(\theta) = \ln \mathsf{E} e^{\theta v}$. Для некоторых распределений эта функция может быть вычислена аналитически, но, как правило, приходится использовать ее сильно состоятельную выборочную оценку

$$\ln \frac{1}{k} \sum_{i=0}^{k-1} e^{\theta v_i} \to \Lambda_V(\theta)\,, \quad k \to \infty\,.$$

2.1 Оценивание эффективной пропускной способности по методу группового среднего

В случае зависимых с.в. $\{v_i\}$, который рассматривается ниже, приходится привлекать специальные статистические методы и имитационное моделирование для оценивания функции $\Lambda_V(\theta)$. Очень важно отметить, что заниженная оценка параметра C, которая нарушает гарантию (1), совершенно неприемлема при работе систем высокой надежности и для таких систем определенное завышение требования (1), безусловно, предпочтительнее.

Наиболее распространенным методом оценивания функции Λ_V в настоящее время является метод группового среднего (*batch-mean*) [9–12]. В этом методе строятся блоки фиксированной длины B,

$$\hat{X}_j = \sum_{i=(j-1)B}^{jB-1} v_i\,, \quad j \geq 1\,,$$

которые (при большом значении B) считаются н.о.р., хотя это утверждение в общем случае является аппроксимацией. При фиксированном размере блока B имеет место сходимость с вероятностью 1 (с в. 1) выборочной оценки

$$\hat{\Lambda}_k(\theta,\, B) := \frac{1}{B} \ln \frac{1}{k} \sum_{i=1}^{k} e^{\theta \hat{X}_i} \to \Lambda_V(\theta,\, B) := \frac{\ln \mathsf{E} e^{\theta \hat{X}}}{B}\,,\ k \to \infty\,,$$

где через $\hat{X}$ обозначен типичный блок. Таким образом, оценка $\hat{\Lambda}_k(\theta,\, B)$ является сильно состоятельной оценкой искомой функции $\Lambda_V(\theta,\, B)$. Соответствующая оценка ЭПС определяется тогда следующим образом:

$$\hat{C}_k(B) = \frac{\hat{\Lambda}_k(\theta^*, B)}{\theta^*}, \tag{4}$$

где, напомним, $\theta^* = -\ln \Gamma / b$. Отметим, что выбор величины блока B является отдельной проблемой, и что, как показано в [7, 13], оценка (4) на самом деле смещена в сторону меньших значений, т. е. ее использование не гарантирует выполнения условия (1). Как было отмечено, это очень нежелательно в контексте анализа высоконадежных систем.

2.2 Регенеративная оценка эффективной пропускной способности

В работах [14, 15] была предложена регенеративная оценка функции $\Lambda_V(\theta^*)$ в тех случаях, когда последовательность $\{v_i\}$ является регенерирующей с моментами регенерации β_k и периодами регенерации $\alpha_k = \beta_{k+1} - \beta_k$, $k \geq 0$. Более точно, фрагменты входного процесса

$$(v_0, \ldots, v_{\beta_1 - 1}), \ldots, (v_{\beta_{k-1}}, \ldots, v_{\beta_k - 1}), \ldots \tag{5}$$

образуют н.о.р. случайные элементы, длины которых $\{\alpha_k\}$ также н.о.р. с.в. Свойства ЭПС, полученной на основе такой оценки, были затем исследованы в работах [7, 8] исходя из рекурсии Линдли для процесса незавершенной нагрузки $\{W(t)\}$ в дискретном времени, т. е. рекурсии

$$W(t+1) = [W(t) + v_t - C]^+\,,\ t = 0,\, 1, \ldots \tag{6}$$

Как показано в [7, 8], построенная регенеративная оценка обладает требуемым свойством консервативности, т. е. обеспечивает гарантию (1) во всех случаях. При данном подходе входящая нагрузка разбивается на *регенеративные блоки*

$$\hat{X}_k := \sum_{i=\beta_k}^{\beta_{k+1}-1} v_i\,,\ k \geq 0\,,\ \beta_0 = 0\,,$$

которые, в отличие от блоков, получаемых методом группового среднего, действительно являются н.о.р. Основные моментные условия формулируются таким образом: дисперсия (типичной) длины блока $\mathsf{E}(\alpha - \mathsf{E}\alpha)^2 := \sigma^2 \in (0,\, \infty)$ и в

некоторой положительной окрестности $(0,\, \theta_0)$ параметра θ справедливо условие $\mathsf{E}e^{\theta\hat{X}} < \infty$ (см. выше), так что с.в. $\hat{X}$ имеет конечные моменты любого порядка. В работе [7] для случая дискретного времени (6) предложена *регенеративная оценка* вида

$$\hat{\Lambda}_k(\theta^*) := \frac{k}{\beta_k} \ln \frac{1}{k} \sum_{i=1}^{k} e^{\theta^* \hat{X}_i} \,,$$

которая строится по k циклам регенерации и при $k \to \infty$ сходится с в. 1 к функции

$$\Lambda_{\mathrm{REG}}(\theta^*) := \frac{1}{\mathsf{E}\alpha} \ln \mathsf{E}e^{\theta^* \hat{X}} \,.$$

(Здесь использован хорошо известный результат теории восстановления: $\beta_k/k \to$ $\to \mathsf{E}\alpha$.) Отметим, что в предельной функции $\Lambda_{\mathrm{REG}}(\theta^*)$ присутствует *средняя длина блока* $\mathsf{E}\alpha$. Таким образом, при данной аппроксимации искомая ЭПС удовлетворяет соотношению:

$$C = \frac{1}{\theta^* \mathsf{E}\alpha} \ln \mathsf{E}e^{\theta^* \hat{X}} \,, \tag{7}$$

а ее оценка, построенная по k циклам регенерации, имеет тот же вид, что и оценка (4):

$$\hat{C}_k(\theta^*) = \frac{\hat{\Lambda}_k(\theta^*)}{\theta^*} \,.$$

Проведенный в работах [7, 8] анализ можно перенести и на рекурсию Линдли, построенную по моментам $\{t_n\}$ прихода заявок в систему. Пусть W_n есть незавершенная работа в системе, которую встречает заявка n, т. е. $W_n = W(t_n^-)$, и пусть $\tau_n = t_{n+1} - t_n$ есть интервал между приходом n-й и $(n+1)$-й заявки, $n \geq 1$, $t_0 = 0$. Тогда

$$W_{n+1} = [W_n + v_n - C\tau_n]^+, \;\; n \geq 0 \,, \;\; W_0 = 0 \,, \tag{8}$$

где v_n есть работа, приносимая в систему заявкой n. (При $C = 1$ получаем классическую рекурсию Линдли, где v_n имеет смысл времени обслуживания заявки n.) Предполагается, что входной процесс является регенерирующим с моментами регенерации $\{\beta_k\}$ (см. (5)), так что интервалы входного процесса на данном цикле, вообще говоря, зависимы. Обозначим через T_i момент начала i-го цикла регенерации входного процесса в непрерывном времени, т. е. $T_i = t_{\beta_i}$, $T_0 = t_{\beta_0} = 0$, через $D_i = T_{i+1} - T_i$ — длину i-го цикла и пусть D есть типичная длина цикла, т. е.

$$D =_{\mathrm{st}} \sum_{k=0}^{\alpha-1} \tau_k = T_1$$

($=_{\text{st}}$ означает стохастическое равенство). Будем предполагать конечность дисперсии длины цикла, $\mathrm{Var}\,[D] < \infty$, $\mathrm{Var}\,[\alpha] < \infty$, в непрерывном и в дискретном времени соответственно. Процессу текущей нагрузки (8) соответствует случайное блуждание

$$Z(n) = \sum_{i=0}^{n-1} (v_i - C\tau_i)\,.$$

Запишем функцию $\Lambda_n(\theta)$ в виде (см. (6)):

$$\Lambda_n(\theta) := \frac{1}{n} \ln \mathsf{E} e^{\theta\, Z(n)} = \Lambda_V^{(n)}(\theta) + \frac{1}{n} \ln \mathsf{E} e^{-\theta t_n C}\,.$$

Затем, как и в работе [8], при анализе второго слагаемого сначала перейдем от шкалы входного потока t_n к шкале длин циклов T_k, а затем используем неравенства:

$$\frac{1}{n} \ln \mathsf{E} e^{-\theta T_{k(n)} C} \geq \frac{1}{n} \ln \mathsf{E} e^{-\theta t_n C} \geq \frac{1}{n} \ln \mathsf{E} e^{-\theta T_{k(n)+1} C}\,,$$

где $k(n) := \max(i : \beta_i \leq n)$ есть число циклов регенерации входного процесса в интервале $[0, n]$. Рассуждая, как в работе [8], можно прийти к такой аппроксимации для величины ЭПС:

$$C = \frac{\ln \mathsf{E} e^{\theta^* \hat{X}}}{\theta^*\, \mathsf{E} D}\,. \tag{9}$$

Таким образом, в данном случае в оценке $\hat{\Lambda}_k(\theta^*)$ вместо выборочной оценки величины $\mathsf{E}\alpha$ надо использовать выборочную оценку средней длины цикла в непрерывном времени $\mathsf{E}D$, т. е. $1/k \sum\limits_{i=1}^{k} D_i$. Если α — число приходов заявок на цикле является моментом остановки по отношению к последовательности н.о.р. интервалов $\{\tau_i\}$, то по тождеству Вальда $\mathsf{E}D = \mathsf{E}\alpha\, \mathsf{E}\tau$ (где τ — типичный интервал), а тогда (9) можно записать в виде:

$$C = \frac{\ln \mathsf{E} e^{\theta^* \hat{X}}}{\theta^*\, \mathsf{E}\alpha \mathsf{E}\tau}\,.$$

Отметим, что во всех рассмотренных случаях пропускная способность C обратно пропорциональна среднему интервалу времени, на котором поступает заданный объем работы, что представляется вполне естественным.

На самом деле, как показывают все проведенные эксперименты, соотношение (7) (а также и (9)) является оценкой сверху искомой ЭПС [7, 8]. Сделаем

замечание, которое до некоторой степени проясняет причину такого переоценивания в случае, когда с.в. $\{v_i\}$ н.о.р. и не зависят от длины цикла α. Тогда по свойству условного математического ожидания

$$\frac{1}{\mathsf{E}\alpha}\ln \mathsf{E}e^{\theta^*\hat{X}} = \frac{1}{\mathsf{E}\alpha}\ln \mathsf{E}\left(\mathsf{E}\left(e^{\theta^*\sum\limits_{i=1}^{\alpha} v_i}|\alpha\right)\right) = \frac{1}{\mathsf{E}\alpha}\ln \mathsf{E}\left[\mathsf{E}e^{\theta^* v}\right]^{\alpha},$$

а по неравенству Йенсена

$$\frac{1}{\mathsf{E}\alpha}\ln \mathsf{E}\left[\mathsf{E}e^{\theta^* v}\right]^{\alpha} \geq \frac{\ln \mathsf{E}e^{\theta^* v}}{\theta^*} = C\,.$$

Это показывает, что, по крайней мере для независимых $\{v_i\}$, рандомизация длины блока в аппроксимации (7) в действительности дает верхнюю границу величины C.

2.3 Регенеративная структура тандемной сети

Для построения регенеративной оценки ключевым является алгоритмическое определение моментов регенерации входящего процесса нагрузки. В данном разделе дано описание этой процедуры для узлов тандемной сети вида $GI/G/1 \to \to \cdots \to \cdot/G/1$, состоящей из $m+1$ односерверных узлов. На вход первого узла в моменты $\{t_n\}$ поступают заявки, а интервалы $\tau_n = t_{n+1} - t_n$ образуют процесс восстановления. Задача состоит в вычислении ЭПС, т. е. скорости обслуживания C в узле $m+1$, гарантирующей выполнение условия (1). Входным потоком в узел $m+1$ является выходной поток узла m. Покажем, что этот поток является регенеративным для рассматриваемого класса сетей. Обозначим через $t_n^{(j)}$ момент прихода n-й заявки в узел $j = 1, \ldots, m$. В частности, $t_n^{(1)} = t_n$ есть момент прихода заявки n в первый узел, а $t_n^{(m+1)}$ — момент ее прихода в узел $m+1$ (ухода из узла m). Для заявки n в каждом узле j пусть $S_n^{(j)}$ есть время ее обслуживания, $W_n^{(j)}$ — время ее ожидания и пусть $\tau_n^{(j)} := t_{n+1}^{(j)} - t_n^{(j)}$ — интервал между приходом завок n и $n+1$ в узел j. Предполагается, что для каждого j с.в. $\{S_n^{(j)},\, n \geq 1\}$ н.о.р. Тогда для каждого узла $j = 1, \ldots, m$ справедлива рекурсия Линдли:

$$W_{n+1}^{(j)} = (W_n^{(j)} + S_n^{(j)} - (t_{n+1}^{(j)} - t_n^{(j)}))^+,\ \ n \geq 0\,.$$

Подчеркнем, что при анализе узлов $j = 1, \ldots, m$ рассматриваются времена обслуживания при условии, что скорость обслуживания в каждом таком узле уже задана и равна 1 (тогда приносимая заявкой работа равна времени ее обслуживания). Однако процесс времени ожидания в узле $m+1$ описывается уже рекурсией Линдли вида (8), т. е.

$$W_{n+1}^{(m+1)} = \left[W_n^{(m+1)} + v_n - C\tau_n^{(m+1)}\right]^+,\ \ n \geq 0\,,$$

где v_n — величина работы, приносимая заявкой n, а C — искомая ЭПС, для оценивания которой будет использована аппроксимация (9). Момент прихода заявки $n+1$ в узел j вычисляется по формуле:

$$t_{n+1}^{(j)} = t_n^{(j-1)} + W_n^{(j-1)} + S_n^{(j-1)}\,,\ \ j = 1, \ldots, m+1\,.$$

Момент регенерации входного потока в узел $m+1$ определяется как момент прихода заявки n, такой что

$$W_n^{(1)} = W_n^{(2)} = \cdots = W_n^{(m)} = 0\,, \tag{10}$$

т. е. заявки, которая *не ожидала обслуживания ни в одном предшествующем узле* $j = 1, \ldots, m$. На самом деле данное условие определяет так называемую *квазирегенерацию*, когда зависимость между циклами регенерации не исключается, но является, как правило, весьма слабой. Таким образом, моменты (квази)регенерации определяются как моменты $T_n := t_n^{(m+1)}$, для которых выполнено условие (10). (В экспериментах, описанных в следующем разделе, под регенерацией понимается квазирегенерация.) Связь классической регенерации, квазирегенерации и занимающей промежуточное положение *слабой регенерации* подробно обсуждается в работах [16–18].

3 Результаты численных экспериментов

В данном разделе дано построение регенеративной оценки ЭПС узла $m+1$ в тандемой сети на основе формулы (9), а также приведены результаты тестирования зависимости качества этой оценки от числа узлов и вида распределения времени обслуживания в узлах $j = 1, \ldots, m$. Интенсивность входного потока λ в узел 1 и интенсивности обслуживания μ_j в каждом узле $j = 1, \ldots, m$ подобраны так, что коэффициент загрузки $\rho_j := \lambda/\mu_j < 1$. Эти условия обеспечивают положительную возвратность регенерирующего входного потока в узел $m+1$, т. е. конечность средней длины его цикла регенерации [19]. Затем строится оценка $\hat{C}$ искомой ЭПС, удовлетворяющей условию (1).

Напомним, что времена обслуживания S_j во всех узлах $j = 1, \ldots, m$, а также работа v_n должны удовлетворять условиям $\mathsf{E}e^{\theta^* S_j} < \infty$, $\mathsf{E}e^{\theta^* \hat{X}} < \infty$. Ниже рассматриваются показательные времена обслуживания, времена обслуживания, распределенные по закону Вейбулла с легким хвостом, а также по (усеченному) закону Парето.

Пример 1. Пусть $b = 10$, $\Gamma = 0{,}01$, тогда $\theta^* = 0{,}460517$. Пусть работа v_i задается как

$$v_i = \frac{1}{i}\sum_{k=1}^{i} \eta_k\,, \tag{11}$$

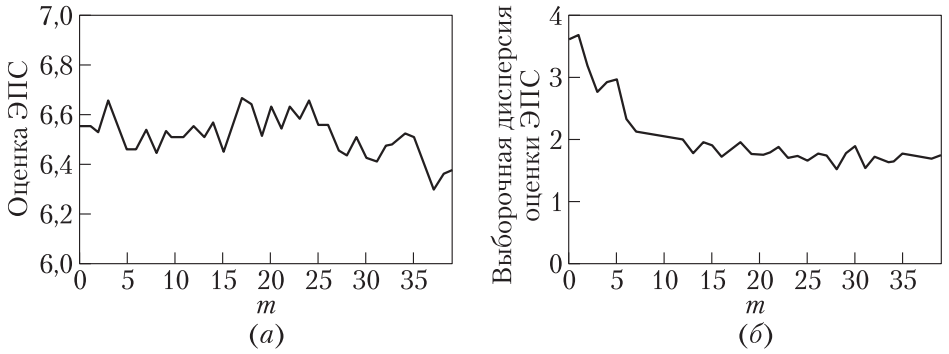


Рис. 1 Пример 1: регенеративная оценка ЭПС $\hat{C}$ (*а*) и выборочная дисперсия оценки ЭПС $\hat{C}$ (*б*) в тандеме из m узлов

где н.о.р. с.в. $\{\eta_k\}$ имеют показательное распределение $F_\eta(x) = 1 - e^{-0{,}4x}$, $x \geq 0$, а i — номер заявки на цикле квазирегенерации, $1 \leq i \leq \alpha$. Времена обслуживания S_j на всех узлах $j = 1, \ldots, m$ распределены экспоненциально с одним и тем же параметром $\mu_j = 1$.

Интервалы τ_i между приходами заявок в первый узел распределены показательно с параметром $\lambda = 0{,}3$, так что $\rho = 0{,}3 < 1$. Поскольку все $\mu_j = 1$, то интенсивность входного потока в каждый узел $j = 1, \ldots, m$ также равна $\lambda_j = 0{,}3$ и поэтому коэффициент загрузки в каждом из этих узлов $\rho_j = 0{,}3 < 1$ [19].

Как показывает рис. 1, *а*, с ростом числа узлов m значение оценки ЭПС меняется незначительно, а рис. 1, *б* показывает, что оценка дисперсии быстро стабилизируется. Например, для $m = 15$ дисперсия $\mathrm{Var}\big[\hat{C}\big] = 1{,}76694$, а для $m = 35$ $\mathrm{Var}\big[\hat{C}\big] = 1{,}63752$. Было проведено сравнение оценки вероятности переполнения $\hat{\Gamma}$ в системе, где в качестве C использовалась оценка $\hat{C}$, с заданным значением $\Gamma = 0{,}01$. Вычисление $\hat{C}$ проводилось для $k = 1000$ циклов регенерации, а для выборочной дисперсии использовалась 1000 данных.

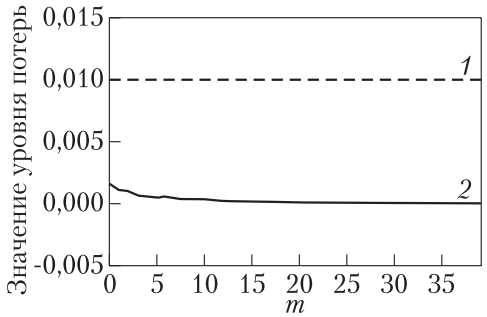


Рис. 2 Пример 1: оценка вероятности переполнения P_b при $\Gamma = 0{,}01$ (*1* — заданный уровень; *2* — оценка)

Из рис. 2 видно, что при любом (рассмотренном) числе узлов оценка стационарной вероятности переполнения $\mathsf{P}(W > b)$ оказывается меньше заданного уровня надежности Γ, что подтверждает консервативность регенеративной оценки ЭПС.

Следует отметить, что с ростом числа узлов также растет и переоценивание, т. е. увеличение запаса требуемой мощности. Например, для 10 узлов $\hat{\Gamma} = 3{,}93959 \cdot 10^{-4}$, а для 35 узлов $\hat{\Gamma} = 5{,}66406 \cdot 10^{-5}$ (см. рис. 3). Однако для систем высокой надежности консервативность регенеративной оценки является скорее преимуществом, тогда как заниженная оценка (которую дает метод группового среднего [7]) может привести к недопустимому нарушению QoS.

Пример 2. Времена обслуживания в каждом (предшествующем) узле — независимые показательные с.в. с параметром $\mu = 1$, а интервалы входного потока в первый узел распределены показательно с параметром $\lambda = 0{,}3$, так что $\rho = 0{,}3 < 1$. Зависимость между с.в. $\{v_i\}$ имеет тот же вид (11), но теперь н.о.р. с.в. $\{\eta_k\}$ имеют распределение Вейбулла с легким хвостом:

$$F_\eta(x) = 1 - e^{-3x^4}, \quad x \geq 0\,. \tag{12}$$

Кроме того, использованы параметры $b = 30$, $\Gamma = 10^{-5}$, так что $\theta^* = 0{,}383764$. Вычисление оценки $\hat{C}$ проводилось для $k = 1000$ циклов регенерации, а для ее выборочной дисперсии использовалась 1000 данных.

На рис. 3, *а* показано, как меняется значение оценки $\hat{C}$ с ростом числа узлов. В этом случае с ростом числа узлов растет и дисперсия оценки $\hat{C}$ (рис. 3, *б*), однако оценка остается консервативной и гарантирует заданный уровень QoS: во всех экспериментах величина $\Delta := (\Gamma - \hat{\Gamma})/\Gamma > 0$. В частности, было подсчитано, что для двух узлов $\hat{\Gamma} = 2{,}95426 \cdot 10^{-6}$ и $\Delta = 0{,}704574$, а для пяти узлов $\hat{\Gamma} = 4{,}9718 \cdot 10^{-7}$ и $\Delta = 0{,}950282$, т. е. переоценивание (по отношению к заданному уровню Γ) с ростом числа узлов растет, как и в примере 1.

Пример 3. Времена обслуживания имеют показательное распределение, но с разными параметрами μ_i, $i = 1, \ldots, m$. При этом дисперсия длин циклов

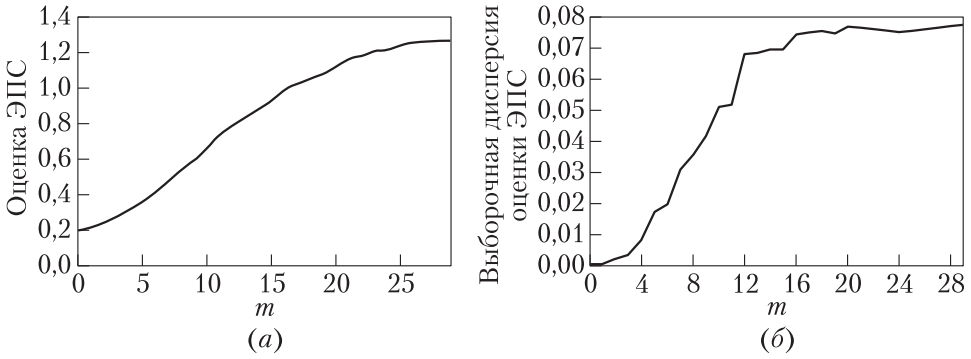


Рис. 3 Пример 2: регенративная оценка ЭПС $\hat{C}$ (*а*) и выборочная дисперсия оценки ЭПС $\hat{C}$ (*б*) в тандеме из m узлов

регенерации больше, чем в предыдущих примерах. Однако, как видно из табл. 1, величина дисперсии оценки ЭПС быстро стабилизируется с ростом числа узлов. В табл. 1 приведены результаты оценивания при $\lambda = 0{,}3$, $\Gamma = 10^{-8}$, $b = 40$ (поэтому $\theta^* = 0{,}460517$), а параметры μ_i выбраны так, что выполнено условие стационарности $\rho = \lambda/\mu_i < 1$, н.о.р. с.в. $\{\eta_k\}$ распределены по закону Вейбулла (12), а с.в. $\{v_i\}$ формируются с помощью (11). Рассматривается сеть, содержащая $m = 2, 4, \ldots, 12$ узлов (первый столбец табл. 1), причем для каждого m в качестве параметров μ_i, $i = 1, \ldots, m$, выбираются первые m значений из набора $\{0.8;\ 0.7;\ 0.5;\ 0.9;\ 0.5;\ 0.8;\ 0.6;\ 0.9;\ 1;\ 1;\ 0.8;\ 0.7\}$. Число циклов регенерации для вычисления оценки $\hat{C}$ равно $k = 3000$, а величина выборки для вычисления дисперсии $\hat{C}$ равна 1000 данных.

В данном примере выбор различных параметров μ_i привел к значительному увеличению дисперсии длин циклов α и D, однако дисперсия оценки $\hat{C}$ стабилизируется с ростом числа узлов достаточно быстро.

Таблица 1 Оценивание ЭПС при различных μ_i

m	$\hat{\alpha}$	$\hat{D}$	$\mathrm{Var}\,[\hat{\alpha}]$	$\mathrm{V}[\hat{D}]$	C	$\mathrm{Var}[\hat{C}]$
2	2,2317	7,4601	5,75	36,51	0,59649	0,07149
4	4,4407	14,7588	42,86	284,89	2,21465	0,29624
6	8,0033	26,4823	152,24	1149,87	2,32183	0,26295
8	10,5023	35,1061	256,80	2015,86	2,35188	0,19200
10	11,9795	40,0484	307,94	2506,86	2,31088	0,19374
12	14,0427	46,6333	428,39	3471,55	2,24337	0,17066

Пример 4. Распределения с *тяжелым хвостом* широко используются для моделирования современного интернет-трафика [20], однако при оценивании ЭПС (в силу моментных требований) приходится использовать усеченные распределения такого вида. Пусть времена обслуживания S_j на узлах j имеют распределение Вейбулла с параметрами $\gamma = 2$, $c = 3$, $\lambda = 0{,}4$ (тогда $\rho = 0{,}44$), а поступающая в узел $m + 1$ на цикле нагрузка $\{v_i\}$ задается соотношением (11), где н.о.р. с.в. $\{\eta_k\}$ имеют усеченное распределение Парето вида

$$F_\eta(x) = \begin{cases} 1 - x^{-4}, & 1 \le x \le L\,; \\ 1\,, & x \ge L\,, \end{cases}$$

где L — некоторая постоянная. Были использованы параметры $b = 30$, $\Gamma = 10^{-5}$, что дает $\theta^* = 0{,}383764$. Число циклов регенерации равно 5000, а размер выборки равен 1000.

Результаты оценивания приведены в табл. 2. В отличие от примера 3, дисперсия длин циклов α и D при увеличении числа узлов меняется незначительно, что, по-видимому, связано с использованием распределения Вейбулла. Например,

Таблица 2 Оценивание ЭПС, когда η_k имеют усеченное распределение Парето

m	$L = 10$		$L = 20$	
	$\hat{C}$	$\mathrm{Var}\big[\hat{C}\big]$	$\hat{C}$	$\mathrm{Var}\big[\hat{C}\big]$
2	0,491	0,0033	0,546	0,1181
4	0,528	0,0096	0,583	0,1175
6	0,561	0,0115	0,623	0,1361
8	0,598	0,0148	0,658	0,1495
10	0,629	0,0270	0,705	0,1611
12	0,652	0,0335	0,736	0,1538
14	0,688	0,0422	0,755	0,1597
16	0,721	0,0477	0,779	0,1428

для двух узлов эксперимент показал, что оценки $\mathrm{Var}\,[\hat{\alpha}] = 0{,}375574$, $\mathrm{Var}\big[\hat{D}\big] = 11{,}1578$, а для 16 узлов $\mathrm{Var}\,[\hat{\alpha}] = 1{,}44008$, $\mathrm{Var}\big[\hat{D}\big] = 13{,}3772$. Поведение оценки ЭПС аналогично: с ростом числа узлов растет величина переоценивания ЭПС, а дисперсия оценки остается стабильной.

4 Заключение

В данной статье продолжены исследования свойств регенеративной оценки ЭПС узла инфокоммуникационной сети. Ранее этот вопрос был исследован для отдельного узла с входным регенерирующим процессом в работах [7, 8]. Рассматриваемое обобщение является наиболее естественным для ациклических сетей, в которых входной процесс восстановления, проходя через узлы сети, оказывается положительно возвратным регенерирующим процессом (при условии, что в узлах выполнено условие стационарности). С использованием результатов теории больших уклонений в данной работе предложена аппроксимация величины ЭПС по аналогии с той, что получена в работах [7, 8]. Качество полученной на основе этой аппроксимации оценки проверено моделированием для ряда тандемных сетей с растущим числом узлов. Для времени обслуживания и величины нагрузки использовались распределение Вейбулла с легким хвостом, усеченное распределение Парето, а также экспоненциальное распределение, а число узлов в сети варьировалось от 2 до 40. Показано, что полученная оценка ЭПС во всех случаях обеспечивает гарантию качества сервиса: оценка вероятности P_b превышения процессом нагрузки заданного уровня b всегда меньше заданного значения Γ. Этот результат может быть использован при выборе величины ЭПС для узлов инфокоммуникационных систем высокой надежности.

Обсудим ряд вопросов, возникающих в связи с проведенным исследованием. Использовать на практике моменты квазирегенерации для оценивания гораздо

эффективнее, чем классическую регенерацию, которая порождается заявками, приходящими в полностью пустую сеть. Моменты классической регенерации в современных сетях либо невозможны, либо слишком редки, чтобы быть полезными для оценивания требуемых параметров за приемлемое время моделирования. Моменты квазирегенерации также являются более частыми, чем моменты *слабой регенерации*, при которой существует зависимость между соседними циклами регенерации, но длины циклов остаются независимыми. (Эти вопросы подробно обсуждаются в работах [17–19].) Отметим также наличие вложенных последовательностей квазирегенераций: заявка, порождающая квазирегенерацию узла k, также порождает квазирегенерацию подсети, содержащей узлы $1, \ldots, k-1$. Наличие таких вложенных подпоследовательностей может быть использовано для сокращения дисперсии при доверительном оценивании [21].

Коснемся возможных причин полученных положительных результатов в сетях с большим числом узлов. Заметим, что в рассматриваемой тандемной сети односерверных узлов входной процесс в последующий узел является комбинацией процессов восстановления, порожденных временами обслуживания на периодах занятости предшествующего узла и следующими за ними периодами простоя. Это в определенной мере должно ограничивать влияние роста длины циклов (квази)регенерации (порождаемое ростом длины сети) на дисперсию оценки $\hat{C}$. Заметим, что с ростом величины нагрузки длина периодов простоя должна убывать, и можно ожидать, что входной процесс в искомом узле будет все более определяться процессом восстановления (временами обслуживания) предшествующего узла. Однако увеличение нагрузки приводит к существенному росту длины циклов квазирегенерации. Укажем также на возможность использования более простого, чем (10), условие квазирегенерации $W_n^{(m)} = 0$, при котором моментом квазирегенерации является приход заявки, не ожидающей обслуживания в узле m.

Ввиду формы зависимости (11) с.в. v_i с ростом i сближается (в силу закона больших чисел) со средним значением $\mathsf{E}\eta$, когда растет длина цикла α. Это можно было бы считать одним из факторов стабилизации дисперсии оценки $\hat{C}$ с ростом длины сети, однако такой эффект наблюдался и при других видах зависимости.

Отметим, что качество оценивания может значительно ухудшиться в произвольной ациклической сети, где могут объединяться несколько различных входных процессов и присутствовать многосерверные узлы.

В системах высокой надежности переоценивание предпочтительнее, но приемлемая величина переоценивания Δ все же зависит от конкретной ситуации. Предварительные эксперименты показали, что даже малое (относительное) уменьшение величины C может нарушить гарантию (1), так что получаемое переоценивание может быть экономически вполне оправдано. Однако этот вопрос требует дальнейшего изучения.

Поскольку значения Γ, как правило, достаточно малы, то серьезной вычислительной проблемой является построение надежной оценки вероятности P_b

при проверке качества оценки $\hat{C}$. Хотя для величины $\Gamma = 10^{-2}$ (см. пример 1) стандартный метод Монте-Карло позволяет построить оценку $\hat{\Gamma}$ за приемлемое время моделирования, однако оценивание, например, вероятности порядка 10^{-8} даже с использованием вычислительных возможностей кластера КарНЦ РАН (содержащего 10 вычислительных узлов на базе четырехъядерных процессоров Intel Xeon 5430 2,66 GHz) потребовало около 24 ч. Таким образом, для эффективного оценивания вероятности P_b необходимо применять технику ускоренного моделирования редких событий, в частности метод расщепления (см., например, [22–24]).

Все затронутые выше вопросы требуют дальнейшего исследования.

Литература

1. *Kelly F.* Notes on effective bandwiths // Stochastic networks: Theory and applications / Eds. F. P. Kelly, S. Zachary, I. B. Ziedins. Royal Statistical Society lecture notes ser., 4. — Oxford: Oxford University Press, 1996. P. 141–168.
2. *Lewis J. T., Russell R.* An introduction to large deviation for teletraffic engineers. DIAS Technical Report DIAS-STP 97-16, 1997.
3. *Crosby S., Leslie I., Huggard M., Lewis J. T., McGurk B., Russel R.* Predicting bandwidth requirements of ATM and Ethernet traffic // 13th IEE UK Teletraffic Symposium Proceedings. — Glasgow, U.K., 1996. P. 1–45.
4. *Glynn P. W., Whitt W.* Logarithmic asymptotics for steady-state tail probabilities in a single-server queue // J. Appl. Probab., 1994. Vol. 31. P. 131–156.
5. *Ganesh A., O'Connell N., Wischik D.* Big queues. — Berlin: Springer-Verlag, 2004. 260 p.
6. *Бородина А. В., Дюденко И. С., Морозов Е. В.* Ускоренное оценивание вероятности переполнения регенеративных систем обслуживания // Обозрение прикладной и промышленной математики, 2009. Т. 16. Вып. 4. С. 577–593.
7. *Бородина А. В., Морозов Е. В.* Сравнение двух оценок эффективной пропускной способности системы обслуживания // Труды Карельского научного центра РАН, 2012. Вып. 6. С. 8–17.
8. *Бородина А. В., Морозов Е. В.* Об оценивании эффективной пропускной способности в системе с регенеративным входным процессом // Информатика и её применения, 2013. Т. 7. Вып. 2. С. 26–33.
9. *Schmeiser B.* Batch size effects in the analysis of simulation output // Oper. Res., 1982. Vol. 30. P. 556–568.
10. *Song W. T.* On the estimation of optimal batch sizes in the analysis of simulation output // Eur. J. Oper. Res., 1996. Vol. 88. No. 2. P. 304–319.
11. *Song W. T., Mingchang Ch.* Implementable mse-optimal dynamic partial-overlapping batch means estimators for steady-state simulations // 2008 Winter Simulation Conference Proceedings, 2008. P. 426–435.
12. *Dyudenko I., Morozov E., Pagano M., Sandmann W.* Comparative study of effective bandwidth estimators: Batch means and regenerative cycles // 6th St. Petersburg Workshop on Simulation Proceedings. — St. Petersburg: VVM com. Ltd., 2009. Vol. II. P. 1003–1007.

13. *Rabinovitch P.* Statistical estimation of effective bandwidth. — Cambridge: University of Cambridge, 2000. M.Sc. Thesis. P. 1–75.
14. *Vorobieva I., Morozov E., Pagano M., Procissi G.* A new regenerative estimator for effective bandwidth prediction // AMICT'2007 Proceedings. — Petrozavodsk: Petrozavodsk State University, 2008. Vol. 9. P. 175–187.
15. *Dyudenko I., Morozov E., Pagano M.* Regenerative estimator for effective bandwidth // Mathematical methods for analysis and optimization of information telecommunication networks: Proceedings of the International Conference. — Minsk: Belarusian State University, 2009. P. 58–60.
16. *Morozov E., Aminova I.* Steady-state simulation of some weak regenerative networks // Eur. Trans. Telecommunications (ETT), 2002. Vol. 13. No. 4. P. 409–418.
17. *Belyy A. V., Aminova I. V.* Queueing network simulation based on quasi-weak regeneration // Inform. Proc., 2002. Vol. 2. No. 2. P. 146–148.
18. *Bodyonov D., Morozov E.* Regenerative simulation of a long range dependent process in a tandem network // 5th St. Petersburg Workshop (International) on Simulation Proceedings. — St. Petersburg: NII Chemistry St. Petersburg University Publs., 2005. P. 169–173.
19. *Morozov E.* Weak regeneration in modeling of queueing processes // Queueing Syst., 2004. Vol. 46. No. 3–4. P. 295–315.
20. *Park K., Willinger W.* Self-similar network traffic and performance evaluation. — New York, NY, USA: John Wiley & Sons, 2000. 576 p.
21. *Andradottir S., Calvin J., Glynn P. W.* Accelereted regeneration for Markov chain simulation // Probab. Eng. Inform. Sci., 1995. Vol. 9. P. 497–523.
22. *Glasserman P., Heidelberger P., Shahabuddin P., Zajic T.* A look at multilevel splitting // Monte Carlo and quasi Monte Carlo methods / Ed. H. Niederreiter. Lecture notes in statistics ser. — Berlin: Springer-Verlag, 1996. Vol. 127. P. 99–108.
23. *Glasserman P., Heidelberger P., Shahabuddin P., Zajic T.* Splitting for rare event simulation: Analysis of simple cases // 1996 Winter Simulation Conference Proceedings. — San Diego, CA, USA: Academic Press, 1996. P. 302–308.
24. *Бородина А. В., Морозов Е. В.* Ускоренное регенеративное моделирование вероятности перегрузки односерверной очереди // Обозрение прикладной и промышленной математики, 2007. Т. 14. Вып. 3. С. 385–397.

Поступила в редакцию 22.03.14

ESTIMATION OF THE EFFECTIVE BANDWIDTH OF A NODE IN AN INFO-COMMUNICATION TANDEM NETWORK

***A. V. Borodina*[1,2] *and E. V. Morozov*[1,2]**

[1]Institute of Applied Mathematical Research, Karelian Research Center, Russian Academy of Sciences, 11 Pushkinskaya Str., Petrozavodsk 185910, Republic of Karelia, Russian Federation
[2]Petrozavodsk State University, 33 Lenin Str., Petrozavodsk 185910, Republic of Karelia, Russian Federation

Abstract: The properties of the effective bandwidth (EB) regenerative estimate of a communication node in the tandem network are investigated. This problem has been studied earlier for a separate node with regenerative input. This setting is natural for acyclic networks because the input renewal process becomes positive recurrent regenerative while crossing the nodes of such a network (under the steady-state condition). Based on the theory of large deviations results, an approximation of the EB is proposed which quality is verified by simulation of a few tandem networks. The Weibull distribution with a light tail, the truncated Pareto distribution, and the exponential distribution are used for service time and the arrived workload, and the number of the nodes is varied from 2 to 40. It is shown that the EB estimator obtained by this approximation ensures the following condition: the overflow probability estimate is always less than the required given value (guarantee of quality of service). This result indicates the possibility to use the proposed approximation for choosing the EB values in the nodes in info-communication highly reliable tandem networks.

Keywords: tandem network; effective bandwidth; regenerative input; quality of service; theory of large deviations; approximation; statistical estimation; simulation

DOI: 10.14357/08696527140203

Acknowledgments

The work was financially supported by the Program of Strategy Development of Petrozavodsk State University in the framework of a set of measures for the development of research activities.

References

1. Kelly, F. 1996. Notes on effective bandwiths. *Stochastic networks: Theory and applications*. Eds. F. P. Kelly, S. Zachary, and I. B. Ziedins. Royal Statistical Society lecture notes ser. Oxford: Oxford University Press. 4:141–168.

2. Lewis, J. T., and R. Russell. 1997. An introduction to large deviation for teletraffic engineers. DIAS Technical Report DIAS-STP 97-16.
3. Crosby, S., I. Leslie, M. Huggard, J. T. Lewis, B. McGurk, and R. Russel. 1996. Predicting bandwidth requirements of ATM and Ethernet traffic. *13th IEE UK Teletraffic Symposium Proceedings*. Glasgow, U.K. 1–45.
4. Glynn, P. W., and W. Whitt. 1994. Logarithmic asymptotics for steady-state tail probabilities in a single-server queue. *J. Appl. Probab.* 31:131–156.
5. Ganesh, A., N. O'Connell, and D. Wischik. 2004. *Big queues*. Berlin: Springer-Verlag. 260 p.
6. Borodina, A. V., I. S. Dyudenko, and E. V. Morozov. 2009. Uskorennoe otsenivanie veroyatnosti perepolneniya regenerativnykh sistem obsluzhivaniya [Speed-up simulation of overflow probability of regenerative queuing systems]. *Obozrenie Prikladnoy i Promyshlennoy Matematiki* [Survey of Applied and Industrial Mathematics] 16(4):577–593.
7. Borodina, A. V., and E. V. Morozov. 2012. Sravnenie dvukh otsenok effektivnoy propusknoy sposobnosti sistemy obsluzhivaniya [Comparison of two estimates of service system effective bandwidth]. *Trudy Karel'skogo Nauchnogo Tsentra RAN* [Proceedings of Karelian Research Center of RAS] 6:8–17.
8. Borodina, A. V., and E. V. Morozov. 2013. Ob otsenivanii effektivnoy propusknoy sposobnosti v sisteme s regenerativnym vkhodnym protsessom [On estimation of the effective bandwidths in a system with regenerative input]. *Informatika i ee Primeneniya — Inform. Appl.*] 7(2):26–33.
9. Schmeiser, B. 1982. Batch size effects in the analysis of simulation output. *Oper. Res.* 30:556–568.
10. Song, W. T. 1996. On the estimation of optimal batch sizes in the analysis of simulation output. *Eur. J. Oper. Res.* 88(2):304–319.
11. Song, W. T., and Ch. Mingchang. 2008. Implementable mse-optimal dynamic partial-overlapping batch means estimators for steady-state simulations. *2008 Winter Simulation Conference Proceedings*. 426–435.
12. Dyudenko, I., E. Morozov, M. Pagano, and W. Sandmann. 2009. Comparative study of effective bandwidth estimators: Batch means and regenerative cycles. *6th St. Petersburg Workshop on Simulation Proceedings*. St. Petersburg. II:1003–1007.
13. Rabinovitch, P. 2000. Statistical estimation of effective bandwidth. University of Cambridge. M.Sc. Thesis. 75 p.
14. Vorobieva, I., E. Morozov, M. Pagano, and G. Procissi. 2008. A new regenerative estimator for effective bandwidth prediction. *AMICT'2007 Proceedings*. Petrozavodsk: Petrozavodsk State University. 9:175–187.
15. Dyudenko, I., E. Morozov, and M. Pagano. 2009. Regenerative estimator for effective bandwidth. *Conference (International) "Mathematical Methods for Analysis and Optimization of Information Telecommunication Networks" Proceedings*. Minsk: Belarusian State University. 58–60.
16. Morozov, E., and I. Aminova. 2002. Steady-state simulation of some weak regenerative networks. *Eur. Trans. Telecommunications (ETT)* 13(4):409–418.
17. Belyy, A. V., and I. V. Aminova. 2002. Queueing network simulation based on quasi-weak regeneration. *Inform. Proc.* 2(2):146–148.

18. Bodyonov, D., and E. Morozov. 2005. Regenerative simulation of a long range dependent process in a tandem network. *5th St. Petersburg Workshop (International) on Simulation Proceedings*. 169–173.
19. Morozov, E. 2004. Weak regeneration in modeling of queueing processes. *Queueing Syst.* 46(3-4):295–315.
20. Park, K., and W. Willinger. 2000. *Self-similar network traffic and performance evaluation*. New York, NY, USA: John Wiley & Sons. 576 p.
21. Andradottir, S. , J. Calvin, and P. W. Glynn. 1995. Accelereted regeneration for Markov chain simulation. *Probab. Eng. Inform. Sci.* 9:497–523.
22. Glasserman, P., P. Heidelberger, P. Shahabuddin, and T. Zajic. 1996. A look at multi-level splitting. In: *Monte Carlo and quasi Monte Carlo methods*. Ed. H. Niederreiter. Lecture notes in statistics ser. Berlin: Springer-Verlag. 127:99–108.
23. Glasserman, P., P. Heidelberger, P. Shahabuddin, and T. Zajic. 1996. Splitting for rare event simulation: Analysis of simple cases. *1996 Winter Simulation Conference Proceedings*. San Diego, CA, USA: Academic Press. 302–308.
24. Borodina, A. V., and E. Morozov. 2007. Uskorennoe regenerativnoe modelirovanie veroyatnosti peregruzki odnoservernoy ocheredi [Speed-up regenerative simulation of the overload probability of a single server queue]. *Obozrenie Prikladnoy i Promyshlennoy Matematiki* [Survey of Applied and Industrial Mathematics] 14(3):385–397.

Received March 22, 2014

Contributors

Borodina Alexandra V. (b. 1980) — Candidate of Science (PhD) in physics and mathematics, scientist, Institute of Applied Mathematical Research, Karelian Research Center, Russian Academy of Sciences, 11 Pushkinskaya Str., Petrozavodsk 185910, Republic of Karelia, Russian Federation; associate professor, Petrozavodsk State University, 33 Lenin Str., Petrozavodsk 185910, Republic of Karelia, Russian Federation; borodina@krc.karelia.ru

Morozov Evsey V. (b. 1947) — Doctor of Science in physics and mathematics, professor, leading scientist, Institute of Applied Mathematical Research, Karelian Research Center, Russian Academy of Sciences, 11 Pushkinskaya Str., Petrozavodsk 185910, Republic of Karelia, Russian Federation; professor, Petrozavodsk State University, 33 Lenin Str., Petrozavodsk 185910, Republic of Karelia, Russian Federation; emorozov@karelia.ru

ИНСТРУМЕНТЫ ДЛЯ СИСТЕМНОЙ ВЕРИФИКАЦИИ РЕКУРРЕНТНОГО ОБРАБОТЧИКА СИГНАЛОВ*

***В. С. Петрухин*[1], *Д. Ю. Степченков*[2], *Н. В. Морозов*[3], *Ю. А. Степченков*[4]**

Аннотация: Рассмотрена процедура выбора и разработки комплекса программных и аппаратных инструментов для проектирования и отладки нетрадиционного цифрового сигнального процессора на базе рекуррентно-динамической потоковой гибридной архитектуры — рекуррентного обработчика сигналов (РОС). Экспериментальный характер отрабатываемой архитектуры РОС и необходимость наличия отработанного управляющего процессора предопределили выбор в качестве элементной базы его реализации ПЛИС (программируемые логические интегральные схемы) семейства Cyclone V фирмы Альтера и, соответственно, среды разработки — Quartus II. Мощные инструменты верификации, входящие в состав этой среды, позволяют сократить время получения готового проекта и существенно уменьшить аппаратные затраты. На основе сравнительного анализа и установленных критериев определен состав и предложена оптимальная структура аппаратных инструментов отладки РОС, позволяющая существенно упростить процесс верификации и отладки РОС в реальном аппаратном окружении.

Ключевые слова: отладочные средства; потоковая архитектура; верификация

DOI: 10.14357/08696527140204

1 Введение

В ИПИ РАН ведутся работы по созданию вычислителя нетрадиционной рекуррентной архитектуры, предназначенного для реализации параллельных вычислений в области цифровой обработки сигналов [1]. Для апробации архитектурных решений и исследования функционирования разрабатываемого РОС он реализуется на ПЛИС. Завершаемым этапом процесса разработки РОС с потоковой архитектурой [2] является верификация проекта РОС на физической модели или системная отладка, т. е. отладка с использованием реальных аппаратных и программных инструментов. Функционирование аппаратных и программных инструментов при этом осуществляется в реальном масштабе времени, что существенно повышает трудоемкость отладки.

*Работа выполнена при частичной финансовой поддержке по Программам фундаментальных исследований ОНИТ РАН за 2013 г. (проект 1.5) и Президиума РАН (проект 16).

[1]Институт проблем информатики Российской академии наук, cokrat2@rambler.ru

[2]Институт проблем информатики Российской академии наук, stepchenkov@mail.ru

[3]Институт проблем информатики Российской академии наук, nmorozov@ipiran.ru

[4]Институт проблем информатики Российской академии наук, ystepchenkov@ipiran.ru

Такой вариант отладки позволяет проверить работоспособность РОС при подаче на него реальных сигналов и в условиях помех, а также устранить нестыковки интерфейсов различных частей РОС и возможные ошибки разводки печатной платы. Довольно часто исследования непосредственно на реальной модели являются единственным средством верификации системы (особенно для столь сложных проектов), поскольку имеются причины, ограничивающие возможности моделирования (симуляции). Во-первых, процесс моделирования может занимать достаточно большое время, так как в модельном времени исследование ведется на несколько порядков медленнее. Во-вторых, достоверность программного моделирования ограничивается соответствием модели входного воздействия реальным условиям.

Можно выделить два основных метода внутрисхемной отладки ПЛИС [3]:

(1) применение встроенных в ПЛИС отладочных инструментов на основе JTAG (Joint Test Action Group);

(2) использование внешнего контрольно-измерительного оборудования: осциллографов смешанных сигналов и логических анализаторов. Использование внешнего оборудования связано со значительными материальными затратами.

В качестве элементной базы реализации РОС выбраны ПЛИС (семейство Cyclone V фирмы Альтера [4]), а в качестве интегрированной среды разработки — система автоматизированного проектирования (САПР) Quartus II, содержащая готовые компоненты для проведения проектирования и отладки систем. Использование таких сред позволяет значительно сократить время разработки и уменьшить временны́е затраты на отладку.

Апробация новых архитектурных решений проводилась в условиях, связанных со значительными временны́ми и финансовыми ограничениями, поэтому основными критериями при выборе состава отладочных средств было максимальное использование стандартных средств отладки, реализованных в Quartus II, и минимальное привлечение дополнительной аппаратуры. В соответствии с выбранными критериями проведен анализ встроенных отладочных средств Quartus II на предмет их использования для отладки устройства РОС.

2 Средства отладки Quartus II

Среда разработки Quartus II имеет следующие встроенные средства отладки [5, 6] (здесь и далее в разд. 2 используется переработанный материал из соответствующих пунктов данных источников в преломлении к поставленным целям и задачам):

– редактор отладочных выводов (SignalProbe Pins);

– редактор интерфейса для внешнего логического анализатора (Logic Analyzer Interface Editor);

– редактор содержимого внутрисхемной памяти (In-System Memory Content Editor);
– встраиваемый логический анализатор SignalTap II (SignalTap II Logic Analyzer);
– редактор контрольных сигналов внутри ПЛИС (In-System Sources and Probes Editor);
– виртуальный JTAG (Virtual JTAG);
– отладочные средства процессора Nios II.

2.1 Отладочные выводы SignalProbe Pins

Отладочные выводы SignalProbe Pins — простейшее средство отладки. Используя этот механизм, можно сделать доступными для наблюдения внутренние сигналы проекта. Для этого они назначаются на не занятые в проекте выводы ПЛИС. Редактор позволяет выбрать соответствие между внутренними сигналами проекта и внешними выводами, назначив их сигналы на не занятые в проекте выводы и соответствующим образом их переименовав.

Простота использования SignalProbe Pins и то, что при использовании отладочных выводов не тратится логический ресурс ПЛИС, являются его основными достоинствами. Нет необходимости в дополнительном оборудовании, благодаря чему сохраняются характеристики пользовательского проекта и отсутствуют крупные дополнительные аппаратные и временны́е затраты. Это делает SignalProbe Pins приемлемым для отладки отдельных блоков схемы. Однако для комплексной отладки РОС он не подходит, потому что не позволяет считывать содержимое памяти. Кроме того, может не хватить неиспользованных выводов.

2.2 Интерфейс для внешнего логического анализатора

При помощи набора управляемых мультиплексоров на вход внешнего логического анализатора отбираются нужные сигналы анализируемого проекта. При этом выходы мультиплексоров назначаются на не занятые в проекте выводы ПЛИС. С помощью JTAG-интерфейса тестируемое устройство подключается к компьютеру. Пользователю предоставляется возможность управления мультиплексорами за счет редактора LAI (Logic Analyzer Interface).

После создания интерфейса для внешнего логического анализатора требуются повторные компиляции проекта, учитывающие изменения в коммутации, и загрузка на ПЛИС. Смена групп отслеживаемых сигналов осуществляется «на лету», что позволяет обойтись в дальнейшем без перезагрузок и перекомпиляций.

Интерфейс LAI может размещаться в микросхемах отладочной платы; внутренние блоки оперативной памяти ПЛИС для его использования не нужны. Интерфейс LAI позволяет решить проблему недостаточного количества внешних выводов, присущую средству SignalProbe Pins, за счет аппаратных затрат

на реализацию интерфейса и привлечения внешнего оборудования. Привлечение дорогостоящего внешнего логического анализатора не соответствует сформулированным критериям выбора средств для отладки РОС. Помимо этого, использование внешнего анализатора может вызвать изменение характеристик пользовательского проекта. Соответственно, использование при отладке РОС интерфейса для внешнего логического анализатора нежелательно.

2.3 Редактор содержимого внутрисхемной памяти

Редактор содержимого внутрисхемной памяти (In-System Memory Content Editor) позволяет анализировать данные, записанные в блоках памяти ПЛИС. Это упрощает отладку РОС, поскольку содержимое блоков памяти РОС модифицируется в процессе работы. Симулятор пакета Quartus II не позволяет решить эту задачу на имитационной модели, поэтому особенно важно решить ее на физической модели. Так как доступ к данным в памяти в процессе работы осуществляется через дополнительный порт, создаваемый редактором содержимого памяти, то возможность анализа доступна только для модулей однопортовой памяти. Связь с ПЛИС осуществляется с помощью JTAG. Соответствующее конфигурирование позволяет редактору содержимого памяти выбирать нужную область памяти с заданными именами и характеристиками, выполнять однократное и циклическое чтение с соответствующим отображением для пользователя и даже редактирование.

Редактор содержимого внутрисхемной памяти — полезный механизм, повышающий скорость и удобство отладки. Его отличительной особенностью является возможность редактирования памяти без дополнительной переконфигурации ПЛИС, отсутствие требования к использованию дополнительной аппаратуры и к ресурсам ПЛИС. Использование такого редактора при отладке РОС перспективно.

2.4 Встраиваемый логический анализатор SignalTap II

Встраиваемый логический анализатор SignalTap II позволяет:

- управлять записью в память на логическом ядре ПЛИС;
- использовать компьютер в качестве средства отображения и анализа ПЛИС;
- иметь память на блоках встроенного оперативного запоминающего устройства (ОЗУ) для записи отсчетов в реальном масштабе времени.

Подключение к внутренним сигналам и выводам микросхемы ПЛИС осуществляется стандартными средствами САПР и вносит минимальные искажения в наблюдаемые сигналы. К SignalTap II, представляющему собой параметризируемую мегафункцию, возможен доступ как с помощью редактора параметризируемых модулей MegaWizard, так и через специализированный пользовательский интерфейс. Соответственно, возможности SignalTap II чрезвычайно широки. Он позволяет выбирать сигналы проекта для наблюдения, осуществлять запись

логических состояний этих сигналов, выполнять совместную работу с САПР Quartus II, подключаясь к ней через JTAG-интерфейс, захватывать наблюдаемые сигналы в реальном времени на частотах свыше 300 МГц.

SignalTap II имеет два режима формирования сигнала захвата: последовательный и режим формирования условий на основе машины состояний. Первый режим (sequential) позволяет использовать и комбинировать стандартные условия захвата. Второй режим (state-based) дает возможность создавать особые условия захвата сигнала для наблюдения, например: комбинацию нескольких событий, появление некоторого события несколько раз и т. п.

Различные настройки позволяют просмотреть данные, записанные до, во время или после захвата сигнала, что существенно упрощает процесс отладки. К сожалению, наблюдение некоторых внутренних сигналов проекта невозможно.

К основным достоинствам SignalTap II можно отнести возможность событийной отладки, создание сложной системы условий, минимально вносимые искажения в наблюдаемые сигналы, возможность анализа изменений сигналов до, во время и после захвата. Это делает его особенно удобным для верификации проектов на основе ПЛИС в реальном аппаратном окружении; он будет основным средством отладки РОС.

2.5 Редактор внутрисхемных сигналов

Редактор внутрисхемных сигналов (In-System Sources and Probes Editor) состоит из мегафункции altsource_probe и интерфейсной графической оболочки, которая позволяет контролировать все элементы мегафункции altsource_probe внутри отлаживаемого проекта в реальном времени. Каждый элемент мегафункции altsource_probe дает для отображения исходные выходные порты (источники) и отводы входных портов (пробники). Интерфейсная графическая оболочка показывает все доступные в реальном времени контролируемые элементы мегафункции altsource_probe в проекте и обеспечивает создание виртуального пульта (виртуальных кнопок и виртуальной лицевой панели) отладки разрабатываемого проекта, имитацию внешнего счетчика данных, отображение и редактирование констант во время прогона, механизм для подвода исследуемых сигналов и средства сохранения данных.

Редактор внутрисхемных сигналов состоит из трех панелей:

(1) конфигуратора цепи JTAG, позволяющего разработчику определять устройство программирования, кристалл, файл настроек редактора;

(2) менеджера элементов, отображающего информацию об элементах, позволяющих контролировать данные динамически;

(3) собственно редактора внутрисхемных сигналов, сохраняющего журнал данных для исследуемых элементов схемы, позволяющего модифицировать эти данные и записывать их в кристалл.

Использование механизмов скриптирования Tcl (Tool command language) позволяет придать отображению сигналов удобный вид, что делает его одним из основных средств отладки.

Для работы редактора внутрисхемных сигналов в рассматриваемом случае требуется программное средство САПР Quartus II с набором его инструментальных средств, загрузочный кабель USB-blaster и отладочная плата.

С помощью редактора внутрисхемных сигналов можно создать виртуальный пульт для управления отладкой РОУ. Для этого пользователь формирует набор виртуальных кнопок и индикаторов (ВКИ), используя приложение In Source In Probe. В процессе отладки он может подавать различные сигналы на входы схем РОУ и следить за состоянием выбранных выходов схем. С помощью виртуального пульта можно проводить ручное тестирование отдельных схем РОУ. Редактор внутрисхемных сигналов сам по себе не требует внешних устройств. Особенно эффективно это средство отладки в сочетании с SignalTap II и In-System Memory Content Editor, давая наибольшую свободу в контроле над сигналами и создании виртуальных входов. Все это позволит существенно сократить время верификации проекта.

2.6 Виртуальный JTAG

Интерфейс VJTAG (Virtual JTAG — виртуальный JTAG) создан для обеспечения обмена данными между платой и компьютером через кабель USB-blaster. Отладочная плата Cyclone V содержит встроенный контроллер JTAG, позволяющий программировать ПЛИС непосредственно в схеме. Контроллер JTAG представляет собой конечный автомат: для получения доступа к его сигналам и обеспечения информационного обмена с внутренними устройствами ПЛИС служит мегафункция VJTAG. Управление интерфейсом VJTAG осуществляется программой quartus_stp, входящей в состав САПР Quartus II, при помощи сценариев на языке Tcl. Конечные устройства (контроллеры памяти, устройства управления светодиодами и пр.) обмениваются данными через декодер с VJTAG-мегафункцией [7].

При работе с отладочной схемой, использующей VJTAG, можно выделить три типа фрагментов этой схемы: встроенные в ПЛИС (порождающие некоторые ограничения по использованию мегафункций), обязательные компоненты VJTAG (не допускающие вариаций) и требующие ручного управления и создания пользователем [8].

При создании интерфейса передачи данных VJTAG требуется:

- создать мегафункцию VJTAG (можно использовать sld_virtual_jtag);
- создать логику, ответственную за коммутацию сигналов и декодирование команд;
- написать сценарий на языке Tcl / Tk (Tool kit);
- создать (подключить) конечные логические устройства.

Все это, с одной стороны, обеспечивает гибкость и уникальность отладочной схемы, поскольку позволяет создавать схемы, зависящие от конкретных требований. С другой стороны, это усложняет использование мегафункции, заставляет детально изучать принципы построения JTAG-контроллеров и создания программного обеспечения (ПО) для них. Ввиду необходимости создания дополнительного оборудования и повышенной трудоемкости Virtual JTAG не отвечает критериям выбора отладочных средств и его использование при отладке РОС нецелесообразно.

Дополнительные отладочные средства в Quartus II появляются в случае использования процессора Nios II, отладочные средства которого построены на основе модуля JTAG.

Программные инструменты отладки связываются с отладочным модулем JTAG и предлагают следующие средства: загрузка программы в память; пуск/останов исполнения программы; установка программных и аппаратных точек останова и точек просмотра; анализ регистров и памяти процессора; накопление следов (данных) исполнения программ в реальном времени.

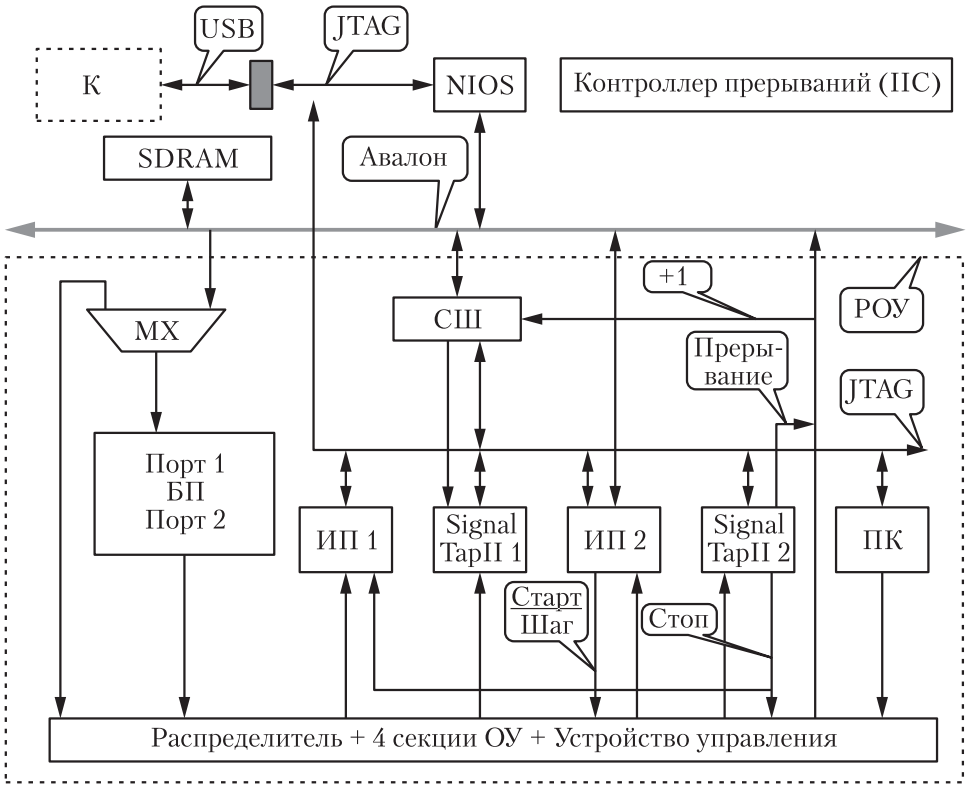
Серьезно расширяет возможности отладки с помощью встроенного логического анализатора SignalTap II плагин Nios II. Он дополняет отладочные средства SignalTap II, разрешая захватывать коды операций процессора Nios II.

В целом процессор Nios II является мощным инструментом отладки, но требующим больших затрат на подготовку и программирование и, к сожалению, не дающий полностью доступа ко всем входам и выходам сигнального процессора, в связи с чем использование процессора Nios II можно рассматривать только как дополнительное.

3 Отладка реккурентного обработчика сигналов

Фирма Альтера не раскрывает протокол взаимодействия персонального компьютера с отладочной платой. Поэтому пользователь вынужден использовать встроенные средства в Quartus или проектировать свои (уникальные) отладочные средства, например на основе виртуального JTAG. Использование компромиссного варианта в виде сочетания встроенных и дополнительно спроектированных средств отладки позволяет оптимизировать временны́е и аппаратные затраты на их реализацию. Структура аппаратных средств РОС с учетом его реализации на основе платы Cyclone V GX FPGA Development Board изображена на рисунке.

Разработанный РОС реализован в виде гибридного двухуровневого варианта с ведущим фон-неймановским процессором на верхнем, управляющем уровне (УУ) и рядом потоковых процессоров на нижнем уровне — рекуррентном операционном устройстве (РОУ). Управляющий уровень, реализованный в виде программы на основе процессора Nios II РОУ, представляет собой VHDL (VHSIC (Very High Speed Integrated Circuits) Hardware Description Language) описания для синтеза на ПЛИС Cyclone V.



Структура аппаратных средств РОС

Обмен данными между этими уровнями осуществляется через буферную память (БП). Для упрощения процесса изложения материала БП отнесена к РОУ. Управляющий уровень состоит из следующих основных модулей: вычислительного ядра Nios II с внутренним контроллером прерываний (IIC — internal interruption controller); системной шины Авалон; памяти процессора Nios II.

В состав РОУ входят следующие функциональные модули: двухпортовая БП; мультиплексор порта 1 записи в БП (МХ); память констант (ПК); распределитель; четыре секции обрабатывающего устройства (ОУ); устройство управления; средства отладки.

Все аппаратные средства РОС взаимодействуют с компьютером (К) через интерфейс USB 2.0. Для обеспечения доступа ко всем функциональным элементам используется стандартный интерфейс JTAG, встроенный логический анализатор SignalTap II и редактор содержимого внутрисхемной памяти.

В свою очередь, дополнительно разработанные средства отладки РОУ состоят из следующих функциональных модулей: счетчика шагов (СШ); модуля

ИП 1 (исходников и пробников) регистров РОУ, модуля ИП 2 (исходников и пробников) виртуальных клавиш и индикаторов, двух анализаторов SignalTap II.

Следует отметить, что аппаратные средства отладки РОУ также привязаны к стандартному интерфейсу диагностики и отладки средств JTAG.

Счетчик шагов исполнения РОУ предназначен для контроля процесса отладки и анализа исполняемой капсулы. Пользователь непосредственно и программа УУ могут изменять содержимое СШ в процессе отладки РОУ.

Модуль ИП 1 обеспечивает возможность изменения и просмотра содержимого регистров ОУ в процессе отладки. Модуль ИП 2 содержит набор ВКИ, предназначенных для управления процессом отладки. Он содержит две группы управляющих виртуальных кнопок:

(1) формирования отладочных команд;

(2) управления работой РОУ.

Пользователь имеет возможность формировать необходимый набор отладочных команд. Для этого набор ВКИ соединен с шиной Авалон, что обеспечивает возможность программного доступа со стороны Nios. Пользователь набирает команду и инициирует ее исполнение на процессоре Nios.

Встроенные средства отладки САПР Quartus II не обеспечивают возможности чтения и редактирования содержимого оперативной двухпортовой памяти. В данном случае БП является двухпортовой. Поэтому функция чтения и записи данных в память осуществляется программным способом с помощью Nios II и модуля ИП 2.

Приложение SignalTap II имеет расширенные возможности и позволяет разработчику создавать и встраивать в РОУ определенное число логических анализаторов, оперативно изменять условия фиксации данных в их памяти и отображать эти данные на экране компьютера. Особенность этого подхода состоит в исследовании поведения внутренних сигналов без использования дополнительных контактов ввода-вывода и какого-либо внешнего оборудования. Пользователь может подсоединять его входы к различным точкам схемы РОУ, задавать условия фиксации сигналов и далее просматривать временные диаграммы. Для этих целей в состав отладочных средств введен логический анализатор SignalTap II.

Управление процессом отладки РОУ пользователь может осуществлять через виртуальный пульт. Для этого, используя редактор внутрисхемных сигналов, пользователь формирует набор ВКИ. В процессе отладки он может подавать различные сигналы на входы схем РОУ и следить за состоянием выбранных выходов схем. С помощью виртуального пульта можно проводить ручное тестирование отдельных схем РОУ.

Архитектура РОУ существенно отличается от традиционной, поэтому процесс отладки носит событийный характер. Для обеспечения возможности настройки и управления процессом отладки РОУ используется формирователь отладочных событий, построенный на основе SignalTap II. Пользователь в приложении

SignalTap II формирует условия отладочных событий, при возникновении которых работа РОУ останавливается и формируется сигнал Прерывание, который инициирует исполнение программы обработки отладочного события в Nios.

4 Заключение

1. Предоставляемые САПР Quartus II инструменты системной отладки удобны в использовании и позволяют существенно упростить процесс верификации проектов на основе ПЛИС в реальном аппаратном окружении.
2. Реализованные в РОС отладочные средства достаточны для поддержки эффективного процесса совместной отладки аппаратных и программных средств.

Литература

1. *Волчек В. Н., Степченков Ю. А., Петрухин В. С., Прокофьев А. А., Зеленов Р. А.* Цифровой сигнальный процессор с нетрадиционной рекуррентной потоковой архитектурой // Проблемы разработки перспективных микро- и наноэлектронных систем-2010: Сб. трудов / Под общ. ред. акад. А. Л. Стемпковского. — М.: ИППМ РАН, 2010. С. 412–417.
2. *Шнейдер А. Ю., Петрухин В. С., Степченков Ю. А.* Принципы построения средств отладки рекуррентного вычислителя // Проблемы разработки перспективных микро- и наноэлектронных систем-2012: Сб. трудов / Под общ. ред. акад. А. Л. Стемпковского. — М.: ИППМ РАН, 2012. С. 133–136.
3. Упрощение отладки ПЛИС Xilinx и Altera // Каталог оборудования 2012–2013. Решения в области контрольно-измерительной аппаратуры. — Tektronix, 2012. С. 386–395. http://www.tehencom.com/Companies/Tektronix/Tektronix_Catalog_2012_Rus.pdf.
4. Cyclone V GX FPGA Development Board: Reference Manual. — Altera Corporation, 2013. http://www.altera.com/literature/manual/rm_cvgx_fpga_dev_board.pdf.
5. *Антонов А., Филиппов А., Золотухо Р.* Средства системной отладки САПР Quartus II // Компоненты и технологии, 2008. № 12(89). С. 43–50.
6. Quartus II Handbook Version 12.0. Vol. 3: Verification. — Altera Corporation, 2013. http://www.altera.com/literature/hb/qts/quartusii_handbook.pdf.
7. *Гребенников А.* Интерфейс VJTAG для отладочной платы DK-START-3C25N // Современная электроника, 2010. № 9. С. 60–63.
8. *Михайлов М., Грушвицкий Р.* Проектирование в условиях временны́х ограничений: отладка проектов (часть 3) // Компоненты и технологии, 2007. № 9(74). С. 133–138.

Поступила в редакцию 31.03.14

SYSTEM VERIFICATION TOOLS FOR RECURRENT SIGNAL PROCESSOR

V. S. Petrukhin, D. Y. Stepchenkov, N. V. Morozov, and Y. A. Stepchenkov

Institute of Informatics Problems, Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation

Abstract: A procedure of selecting and developing a software and hardware suite is studied which is intended for designing and debugging a nontraditional digital signal processor based on the recurrently-dynamic dataflow architecture — the recurrent signal processor (RSP). The experimental character of the developed RSP's architecture as well as a necessity for a finished master processor have predetermined Cyclone V family FPGA (Field Programmable Gate Array) (Altera) as the base of RSP implementation and Quartus II design software for its development. The powerful verification tools contained by Quartus II allow both reducing the time of obtaining the finished design and reducing the hardware expenses essentially. On the basis of the comparative analysis and selected criteria, the tools composition for debugging RSP is determined, and an optimal structure of hardware for debugging RSP is offered that allow for essential simplification of the verification process and for debugging RSP in a real hardware environment.

Keywords: debugging means; dataflow architecture; verification

DOI: 10.14357/08696527140204

Acknowledgments

The research was performed with the partial financial support for Programs of Basic Research of the Department of Nanotechnologies and Information Technologies of the Russian Academy of Sciences for 2013 (project 1.5) and the Presidium of the Russian Academy of Sciences (project 16).

References

1. Volchek, V. N., Yu. A. Stepchenkov, V. S. Petrukhin, A. A. Prokof'ev, and R. A. Zelenov. 2010. Cifrovoy signal'nyy protsessor s netraditsionnoy rekurrentnoy potokovoy arkhitekturoy [Digital signal processor with nontraditional recurrent dataflow architecture]. *Problemy Razrabotki Perspektivnykh Mikro- i Nanoelektronnykh Sistem* [Problems of the Perspective Micro- and Nanoelectronic Systems Development-2010]. Ed. akad. A. L. Stempkovskij. Moscow: IPPM RAN. 412–417.
2. Shnejder, A. Ju., V. S. Petrukhin, and Yu. A. Stepchenkov. 2012. Printsipy postroeniya sredstv otladki rekurrentnogo vychislitelya [Development principles of debugging tools

for the recurrent computing device]. *Problemy razrabotki perspektivnykh mikro- i nanoelektronnykh sistem* [Problems of the perspective micro- and nanoelectronic systems development-2012]. Ed. acad. A. L. Stempkovskij. Moscow: IPPM RAN. 133–136.

3. Tektronix. 2012. Uproshchenie otladki PLIS Xilinx i Altera [Simplification of debugging FPGA Xilinx and Altera]. Katalog oborudovaniya 2012–2013. Resheniya v oblasti kontrol'no-izmeritel'noy apparatury [Equipment 2012–2013 directory. Decisions in the field of control instrumentation]. 386–395. http://www.tehencom.com/Companies/Tektronix/Tektronix_Catalog_2012_Rus.pdf (accessed March 31, 2014).
4. Altera Corporation. 2013. Cyclone V GX FPGA Development Board Reference Manual. URL: http://www.altera.com/literature/manual/rm_cvgx_fpga_dev_board.pdf (accessed March 31, 2014).
5. Antonov, A., A. Filippov, and R. Zolotuho. 2008. Sredstva sistemnoy otladki SAPR Quartus II [System debugging tools of Quartus II CAD]. *Komponenty i Tekhnologii* [Components and Technologies] 12:43–50.
6. Altera Corporation. 2013. Quartus II Handbook Version 12.0. Vol. 3: Verification. URL: http://www.altera.com/literature/hb/qts/quartusii_handbook.pdf (accessed March 31, 2014).
7. Grebennikov, A. 2010. Interfeys VJTAG dlya otladochnoy platy DK-START-3C25N. [VJTAG interface for development board DK-START-3C25N]. *Sovremennaya Elektronika* [The Modern Electronics] 9:60–63.
8. Mihajlov, M., and R. Grushvickij. 2007. Proektirovanie v usloviyakh vremennykh ogranicheniy: Otladka proektov (Chast' 3). [Design in the time restriction conditions: Debugging of projects (part 3)]. *Komponenty i Tekhnologii* [Components and Technologies] 9:133–138.

Received March 31, 2014

Contributors

Petrukhin Vladimir S. (b. 1949) — senior scientist, Institute of Informatics Problems, Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; cokrat2@rambler.ru

Stepchenkov Dmitri Yu. (b. 1973) — senior scientist, Institute of Informatics Problems, Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; stepchenkov@mail.ru

Morozov Nikolai V. (b. 1956) — senior scientist, Institute of Informatics Problems, Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; nmorozov@ipiran.ru

Stepchenkov Yuri A. (b. 1951) — Candidate of Science (PhD) in technology, Head of Department, Institute of Informatics Problems, Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; ystepchenkov@ipiran.ru

СОЗДАНИЕ ВЫСОКОПРОИЗВОДИТЕЛЬНОГО ГЕНЕРАТОРА НАГРУЗКИ ДЛЯ ПРОВЕРКИ СИСТЕМ ВЫСОКОЧАСТОТНОЙ ТОРГОВЛИ*

Д. К. Гурьев[1], *М. А. Гай*[2], *И. Л. Иткин*[3], *А. А. Терентьев*[4]

Аннотация: В связи с существенным ростом числа торговых заявок, вызванным развитием высокочастотной торговли, возникает необходимость и ставится задача тестирования биржевых и брокерских систем в режимах, максимально приближенных к реальным. Для обеспечения качества высоконагруженных трейдинговых систем высокой доступности применяются специализированные инструменты тестирования. Основные требования к таким инструментам — это способность создавать высокие реалистичные нагрузки, используя ограниченную аппаратную базу. В данной статье описывается разработанный генератор нагрузки для тестирования систем автоматизированной торговли. Представлен подход, который обеспечивает высокую производительность. Созданный и описанный инструмент тестирования используется при измерении пропускной способности и времен отклика крупномасштабных биржевых и брокерских платформ, обеспечивающих технологическую инфраструктуру финансовых рынков.

Ключевые слова: автоматизация тестирования; трейдинговые системы; HiVAT

DOI: 10.14357/08696527140205

1 Введение

Высокочастотная электронная торговля финансовыми инструментами (HFT — High Frequency Trading), позволяющая минимизировать временны́е задержки при совершении сделок, выросла в последние годы и составляет в настоящее время около 30% от всего объема торговли акциями в Великобритании и, возможно, более 60% от всего объема торговли акциями в США [1]. В результате этого брокерские и биржевые системы испытывают все бóльшую нагрузку от потока транзакций, генерируемого системами автоматизированной торговли. Операторы трейдинговых платформ, регулирующие органы и участники торгов

*Статья рекомендована к публикации в журнале Программным комитетом конференции “Tools & Methods of Program Analysis” («Инструменты и методы анализа программ», TMPA-2013, 10–12 октября 2013 г., г. Кострома, РФ).

[1]ООО «ИТС-ЭКСПЕРТ», г. Саратов, Dmitry.Guriev@exactprosystems.com

[2]ООО «ИТС-ЭКСПЕРТ», г. Москва, Maria.Gai@exactprosystems.com

[3]Exactpro Systems LLC, США, Калифорния, Iosif.Itkin@exactprosystems.com

[4]ООО «ИТС-ЭКСПЕРТ», г. Саратов, a_a_terentyev@mail.ru

должны быть уверены в надежности программного обеспечения и инфраструктуры торговых площадок [2] в условиях постоянно растущих нагрузок.

В ходе разработки программного обеспечения для выявления максимальной пропускной способности, возможных узких мест и определения проблемных элементов системы применяют методы нагрузочного тестирования. Под нагрузочным тестированием понимается процесс отправки системе большого числа запросов, проверка своевременности и корректности полученных от нее откликов, а также проверка внутреннего состояния системы.

В настоящее время для нагрузочного тестирования программного обеспечения широко используются различные коммерческие и свободно распространяемые генераторы нагрузки. В качестве примера можно привести следующие продукты: Apache JMeter, HP Load Runner, IBM Rational Performance Tester, Borland Silk Performer и др. [3–6]. Основная идея, заложенная в этих продуктах, состоит в создании множества виртуальных пользователей, эмулирующих поведение реальных пользователей для моделирования условий, при которых программа/система будет функционировать в реальности. При имитации и поддержке соединения с большим количеством пользователей и высокой нагрузке у инструментов тестирования могут возникать ограничения производительности, разобранные во второй части данной статьи.

В компании Exactpro Systems LLC разработан инструмент для тестирования высоконагруженных трейдинговых систем, обладающий необходимой производительностью и использованный на практике при проверке некоторых из крупнейших биржевых технологических инфраструктур в Западной Европе [7, 8]. Разработанный инструмент поддерживает протоколы: FIX (Financial Information eXchange) (все версии), ITCH, LSE (London Stock Exchange) Native, Sola Access Information Language (SAIL) & High-Speed Vendor Feed (HSVF), HTTP (HyperText Transfer Protocol), SOAP (Simple Object Access Protocol), а также различные бинарные протоколы трейдинговых систем. Многопротокольность является одной из архитектурных особенностей разработанного инструмента для тестирования, вследствие чего добавление новых протоколов и новых версий уже поддерживаемых протоколов представляет собой относительно малозатратную задачу. В третьей части статьи рассмотрены особенности подготовки данных для нагрузочного тестирования. В четвертой части представлены возможности по настройке инструмента, включая задание профиля нагрузки.

2 Оптимизация процесса создания нагрузки

Общее представление о процессе создания нагрузки дается в ряде работ, в частности в [9]. Генераторы нагрузки делят на основанные на измерениях (measurement-based) и основанные на моделях (model-based) [10]. Основанные на измерениях генераторы удобны для нахождения пропускной способности тестируемой системы и построения зависимостей времен отклика от нагрузки. Генераторы нагрузки, основанные на моделях, направлены на симуляцию

распределения входных данных, максимально приближенную к реальному промышленному использованию системы. В разработанном авторами высокопроизводительном генераторе нагрузки поддерживаются оба описанных варианта. Данные модели используются при создании конфигурационных файлов перед запуском генератора. Таким образом, инструмент может не тратить ресурсы на обработку информации, относящейся к модели, непосредственно во время выполнения тестов.

Генераторы нагрузки подразделяются на работающие по принципам закрытого (closed-cycle) и открытого цикла (open-cycle) [11]. После посылки сообщения исполняемый поток в генераторе закрытого типа дожидается ответа от системы, прежде чем приступить к посылке следующих запросов. Генератор открытого цикла продолжает посылку сообщений, не дожидаясь ответа от тестируемой системы. Большинство инструментов для тестирования веб-приложений являются генераторами закрытого цикла. Это связано с использованием концепции виртуальных пользователей, каждый из которых последовательно выполняет шаги определенного сценария. Генератор закрытого цикла требует существенно большего количества потоков исполнения и переключений между ними в сравнении с генератором открытого цикла. В генераторах закрытого цикла часто обработка исходящих из системы ответов происходит в том же потоке, что и посылка входящих сообщений, дополнительно снижая производительность инструмента и иногда даже влияя на его аккуратность. Таким образом, генераторы открытого цикла требуют меньшей аппаратной базы для создания требуемого уровня нагрузки. Они также не требуют порождения лишних потоков и их синхронизации для производства фиксированного уровня нагрузки. Представленный в статье инструмент работает как генератор открытого цикла.

При разработке инструмента для нагрузочного тестирования рассматривался вопрос о необходимости привязки исполняемых потоков к ядрам процессора для сглаживания распределения входящих в систему сообщений по времени [12]. Авторы пришли к выводу, что миллисекундное разрешение системных таймеров, присутствующее в большинстве современных Linux-систем, достаточно для создания реалистичной трейдинговой нагрузки, а отсутствие привязки к ядрам процессора освобождает некоторый дополнительный объем аппаратных ресурсов на генераторе нагрузки, позволяя централизованно запускать оптимальное число потоков. Разработанный инструмент использует центральный контроллер, позволяя заранее задать в конфигурации количество и состав протокольных соединений, которые будут работать в каждом потоке. Наличие центрального контроллера также позволяет выдавать команду на выполнение скоординированных действий всеми потоками: например, одновременный старт потока сообщений или одновременное отключение установленных соединений.

При тестировании трейдинговой системы генератор нагрузки заменяет большое число систем автоматизированной торговли, использующих множество серверов. Однако аппаратная база, доступная для размещения инструментов тестирования, всегда ограничена соображениями экономии [13]. В условиях про-

должающейся финансовой нестабильности даже крупнейшие финансовые институты работают в режиме максимально возможной оптимизации затрат. Требуется существенно облегчить процесс создания исходящих сообщений. Для оптимизации процесса создания нагрузки необходимо до выполнения теста заготовить шаблоны сообщений, сократив процессорное время на серверах, содержащих инструменты для тестирования. Сходные соображения заложены в генератор нагрузки, созданный разработчиками крупнейшего российского поисковика «Яндекс». Инструмент для тестирования с открытым кодом «Яндекс.Танк» предназначен для генерации огромных объемов сообщений по протоколу HTTP [14]. Высокая производительность «Яндекс.Танк» достигается концентрацией нагрузки в одной сессии и одном потоке, а также использованием заготовленного файла со статическими запросами. Генераторы нагрузки для трейдинговых систем не могут использовать статические данные и обладают рядом других ограничений, рассматриваемых в следующей части.

3 Особенности нагрузочного тестирования торговых систем

В [15] разобраны основные требования к моделированию нагрузки для систем высокочастотной торговли. Логика работы таких систем существенно затрудняет использование статичных, ранее записанных или предопределенных данных. В этом разделе рассматриваются некоторые из особенностей создания нагрузки и подготовки входных данных для тестирования трейдинговых систем.

3.1 Создание сообщений из шаблонов

Вследствие того, что торговля не анонимна, для поддержания сессии сервер должен получать имена существующих пользователей, корректные порядковые номера сообщений, а также времена отправки каждого сообщения. Проведенный анализ показал, что построение сообщений непосредственно перед отправкой с использованием словарей обходится очень дорого с точки зрения используемых системных ресурсов. Поэтому было решено использовать заготовки, в которых порядок полей и набор ключевых значений заданы до начала теста. Например, в FIX-сообщении NewOrderSingle перед моментом отправки будут изменяться всего несколько параметров, которые должны быть уникальны (например, ClOrdID(11) — номер клиентской заявки) и зависеть от текущего времени (ExpireTime(126) — время истечения срока заявки, ExpireDate(432) — день истечения срока заявки). Остальные параметры новой заявки не изменяются. Для поддержания сессии изменяются служебные параметры:

BodyLength(9) — длина сообщения;
SenderCompID(49) — название компании, которая отправила заявку;
TargetCompID(56) — название компании, которой была отправлена заявка;
MsgSeqNum(34) — уникальный номер сообщения;
SendingTime(52) — текущее время;
CheckSum(10) — проверочное число.

В сообщения OrderCancelReplaceRequest и OrderCancelRequest подставляются все необходимые параметры, взятые из сообщения NewOrderSingle, такие как:

OrderID(37) — идентификатор заявки;

Price(44) — цена заявки;

Quantity(53) — размер заявки;

Side(54) — сторона заявки (покупка или продажа);

Symbol(55) — символическое обозначение инструмента.

Предположение, что все значения параметров верны, позволяет существенно экономить время на проверку значений полей и правильности их последовательности в сообщении.

3.2 Воспроизведение ранее записанных данных

Отправка заранее подготовленных записанных данных приводит к искажению теста. При построении данных для тестирования необходимо придерживаться той же пропорции торговых событий, которая наблюдается в реальной жизни. Анализ реальных торгов показывает, что на одну сделку может приходиться более 20 изменений заявок.

Записанные данные представляют собой совокупность новых заявок, их изменения и отмены. Так как заявки отправляются из разных потоков на одну и ту же книгу заявок, они могут приходить на рынок в порядке, отличном от того, каким он был при записи. Даже если время появления заявки на рынке будет отличаться незначительно, это может привести к нежелательным последствиям. Например, к моменту получения рынком заявки другие заявки могут располагаться в книге заявок в порядке, отличном от того, который был при записи, и они могут проторговаться в другой последовательности. Из этого следует, что последующие изменения или отмена заявки будут невозможны, так как заявка проторговалась и была удалена из книги заявок. Такая ситуация влечет за собой сбой в последовательности сценария тестирования, и в дальнейшем будет получен сценарий, отличный от записанного. Таким образом, на рынке будут происходить события, отличные от тех, что были при записи сценария. Например, рынок будет отправлять значительно больше отказов на изменение или отмену заявок. При этом отличия от первоначального сценария могут накапливаться и реальная пропорция торговых событий может сильно отличаться от исходной.

3.3 Использование детерминированных сценариев

Другая возможность организовать тест заключается в подготовке двух групп заявок: к первой группе относятся заявки, о которых заранее известно, что они будут участвовать в сделках (активные заявки); вторая группа состоит из заявок, которые не должны торговаться (пассивные заявки). Однако в этом

случае следует учитывать возможные осложнения со стороны системы мониторинга и контроля рынков (*англ.* Market Surveillance System). Одна из функций этого компонента заключается в том, что он должен в режиме реального времени отслеживать «договорные» заявки, т. е. как раз те заявки, которые должны проторговаться при проведении тестирования, и сообщать о таких событиях службе, следящей за манипулированием ценами на рынке. Очевидно, что такой вариант не подходит для создания сценария тестирования. В связи с этим специалистами компании был создан рандомизированный генератор нагрузки с использованием механизма обратной связи. Рандомизация используется для генерации новой цены при отправке изменения заявки. Для генерации используются три параметра: начальная цена, диапазон изменения цены и шаг изменения цены.

Начальная цена задается в массиве данных для каждого инструмента и для каждой стороны (покупки или продажи). Начальные цены должны удовлетворять следующим условиям:

– цена покупки должна быть меньше цены продажи;

– разница начальных цен продажи и покупки должна составлять около 2%–3% от цены открытия рынка.

Диапазон изменения цены покупки и продажи выбирается таким образом, чтобы цены встречных предложений пересекались, обеспечивая торговлю, и чтобы цены предложений не выходили за 10%-ный барьер — условие, при несоблюдении которого возможна остановка торговли на данном инструменте или на целом сегменте инструментов. Эти параметры позволяют гибко подбирать желаемое среднее соотношение количества сделок и числа изменений, приходящихся в среднем на одну заявку. Чем меньше область пересечения цен покупки и продажи, тем меньше будет сделок и заявка в среднем будет изменяться большее количество раз. Надо заметить, что это отношение нелинейно зависит от интервала пересечения цен. Поскольку цены задаются для каждого инструмента в отдельности, становится возможным настроить разное число сделок на разных инструментах в одном тесте.

Шаг изменения цены задается исходя из конфигурации инструмента. Например, одни инструменты могут торговаться с шагом цены 0,05, другие — с шагом цены 0,10.

Механизм обратной связи необходим, чтобы отслеживать состояние заявки на рынке и обеспечивать возможность изменения или отмены заявки. Заявка может иметь следующие состояния:

New — заявка еще не приняла участие в торговле;
PartFilled — заявка выполнена частично;
Filled — заявка выполнена полностью;
Canceled — заявка отменена клиентом;
Expired — заявка с истекшим сроком действия;
Rejected — заявка отклонена биржей.

Только заявки в состояниях New и PartFilled могут участвовать в торговле. Как только заявка переходит в другое состояние, она перестает быть интересной и информация о ней тут же удаляется.

4 Пример настройки разработанного генератора нагрузки

В этой части статьи подробнее остановимся на нескольких вариантах настройки разработанного авторами генератора нагрузки для тестирования трейдинговых систем на основе протокола FIX [16].

Настройка теста осуществляется посредством четырех типов конфигурационных файлов, которые содержат:

(1) настройку параметров нагрузки;

(2) конфигурацию сессий;

(3) заготовку сообщений;

(4) распределение по сообщениям.

4.1 Формат файла параметров нагрузки

Для настройки параметров нагрузки используется файл, имеющий следующий формат:

```
#Конфигурационный файл с настройками сессий:
CONNECTIONS_CONFIG = fixConnections.cfg
#Указание используемых сессий из файла с сессиями:
CONNECTIONS_RANGE = 1-3, 5, 7-
#Файл с заготовками сообщений:
MESSAGE_TEMPLATES = fixMessageTemplates.dat
#Файл с распределением по сообщениям:
MESSAGE_RATES = messageRates.cfg
#Последовательность действий до начала теста:
INIT_CONFIG = connect(100ms), logon(3s)
#Конфигурация нагрузки:
LOAD_CONFIG = const(1000,5m)
#Задается постоянная нагрузка 1000 сообщений в секунду
#на протяжении пяти минут.
#Количество повторений нагрузочного сценария, заданного
#параметром LOAD_CONFIG:
NUMBER_REPETITIONS = 10
#Последовательность действий после окончания теста:
SHUTDOWN_CONFIG = logout(1s), disconnect(10ms)
#Последовательность действий при внезапном обрыве
#соединения:
ON_RECONNECT_CONFIG = connect(10ms), logon(3s)
```

```
#Флаг на выполнение действий, указанных в
#ON_RECONNECT_CONFIG при обрыве соединения:
HOLD_CONNECTION = 1
#Если значение = 0, действия в ON_RECONNECT_CONFIG
#не выполняются и соединение не восстанавливается.
#Время задержки между авторизацией сессий в миллисекундах
LOGON_INTERVAL = 1000
```

Поскольку у клиентов есть возможность использовать собственные программы для торговли, существует вероятность того, что по какой-то причине, например в случае отправки торговой системой большого объема сообщений, клиентские программы не смогут вовремя прочитать эти данные. Это может негативным образом повлиять на поведение торговой системы. Для тестирования такой возможности в разработанном программном продукте существует специальное ограничение по количеству читаемых данных в секунду для эмуляции медленных клиентов.

4.2 Формат файла конфигурации сессий

Настройки параметров соединений задаются в файле следующего формата:

В секции COMMON задаются общие параметры соединений:

```
[COMMON]
HOST = 10.10.10.10
PORT = 5555
TARGET_COMP_ID = FGW
```

В секции [FIX] задаются уникальные параметры отдельного соединения:

```
[FIX]
SENDER_COMP_ID = LOAD_1
RESET_SEQ_NUM_AFTER_LOGOUT = 0
PARTY_ID = LOAD_1
```

Секция [FIX] должна повторяться столько раз, сколько предполагается использовать соединений. При этом соединения, которым предстоит участвовать в тесте, задаются параметром CONNECTIONS_RANGE в файле с параметрами нагрузки.

4.3 Формат файла заготовок сообщений

Файл содержит массив именованных сообщений-заготовок. Эти заготовки имеют правильный формат и последовательность полей в сообщениях. Некоторые поля будут заменяться корректными данными непосредственно перед отправкой сообщения.

```
Logon
8=FIXT.1.1|9=61|35=A|34=1|49=SenderCompID|56=TargetCompID|98=0|108
=3600|554=password|1137=9|10=135| EOM
```

NewOrderBuy
8=FIXT.1.1|9=199|35=D|34=1|49=SenderCompID|56=TargetCompID|1
=CLIENT|11=ClOrdID|38=200|40=2|44=9.8|54=1|55=Symbol|59=6|60=
20130728-13:34:03.194|432=20130730|528=P|581=3|1138=60000|9303=I|453
=1|448=PartyID|447=D—452=76| 10=047| EOM

NewOrderSell
8=FIXT.1.1|9=199|35=D|34=1|49=SenderCompID|56=TargetCompID|1
=CLIENT|11 =ClOrdID|38=150|40=2|44=10.2|54=2|55=Symbol|59=6|60=
20130728-13:34:03.194|432=20130730|528=P|581=3|1138=60000|9303=I|453
=1|448=PartyID|447=D|452=76|10=047|EOM

Cancel
8=FIXT.1.1|9=134|35=F|34=1|49=SenderCompID|56=TargetCompID|11=
ClOrdID|41=OrigClOrdID|54=1|55=Symbol|60=20130728-13:34:03.178|9303=I|
453=1|448=PartyID|447=D|452=76|10=050|EOM

Replace
8=FIXT.1.1|9=179|35=G|34=1|49=SenderCompID|56=TargetCompID|1=
CLIENT|11=ClOrdID|38=180|40=2|41=OrigClOrdID|54=1|55=Symbol|60=
20130728-13:34:03.178|432=20130730|1138=70000|9303=I|453=1|448=
PartyID|447=D|452=76|10=077|EOM

Logout
8=FIXT.1.1|9=29|35=5|34=111|49=SenderCompID|56=TargetCompID|
10=249|EOM

4.4 Формат файла распределения нагрузки по сообщениям

Файл содержит соотношение количества сообщений, указанных в долях для каждого типа сообщения:
NewOrderBuy = 15
Replace = 50
Cancel = 5

В зависимости от настройки, MESSAGE_SELECTION_ORDER = sequential или random, сообщения будут выбираться или последовательно, или случайным образом.

4.5 Упрощенная схема работы алгоритма

На рис. 1 приведена блок-схема алгоритма по выбору и отправке сообщений. На первом этапе происходит чтение входящих данных. В том случае, если данные есть, происходит анализ полученных ответов и изменение состояния заявок или их параметров. После этого случайным методом или последовательным перебором выбирается новое сообщение. Если был выбран приказ на создание новой заявки, его параметры сохраняются в памяти для дальнейшего использования. В случае выбора сообщения для изменения или отмены заявки, выбирается заявка, для которой нет оставленного без ответа запроса. Ее параметры подставляются в

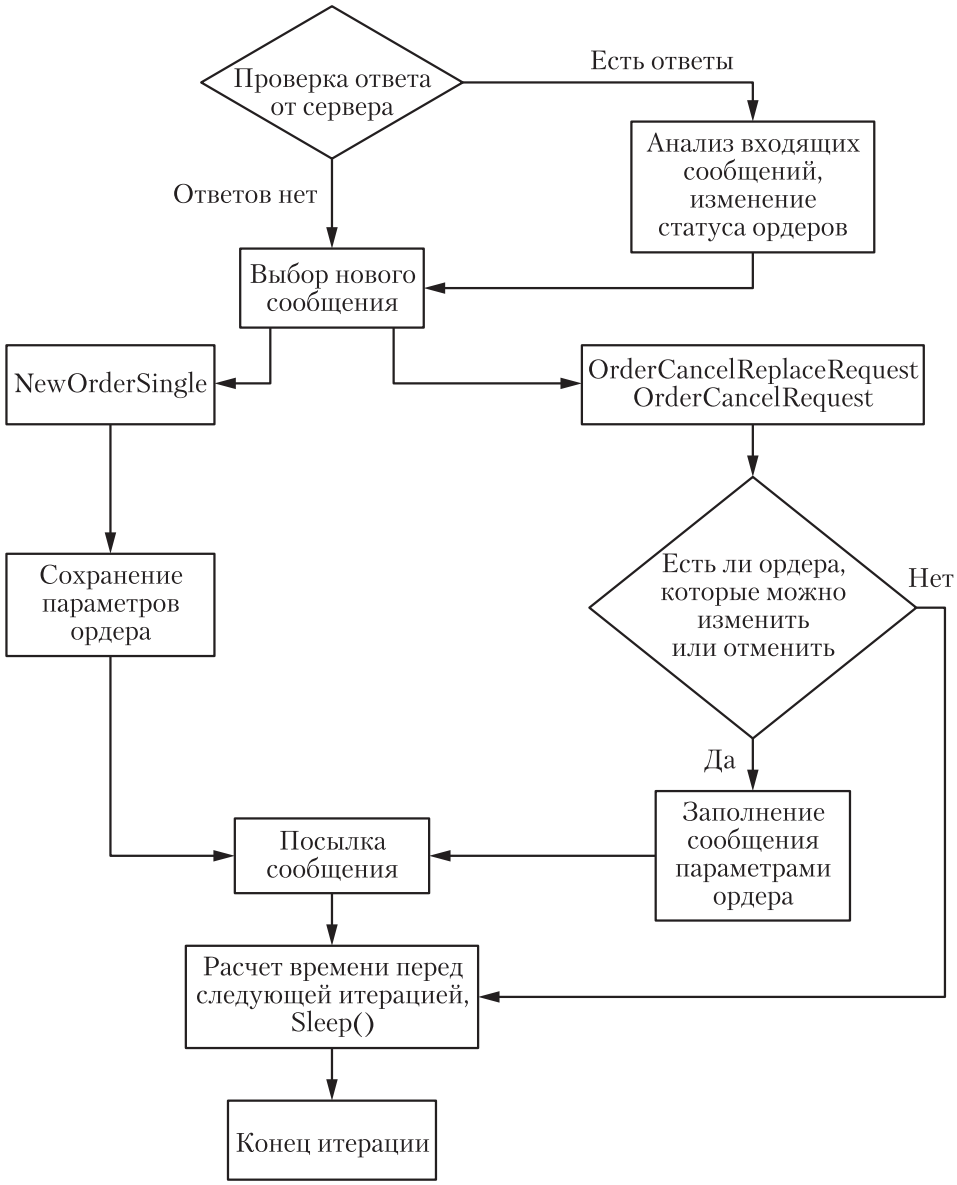


Рис. 1 Блок-схема получения и отправки сообщения

сообщение и посылаются в торговую систему. После этого вычисляется время, прошедшее с начала итерации. Оно сравнивается с расчетным средним временем на одну итерацию, и при необходимости делается пауза. Последовательность «читать–отправить–ждать» позволяет принимать во внимание все последние изменения внутри тестируемой системы и на их основе посылать корректные с функциональной точки зрения сообщения.

Запуск теста с максимальной нагрузкой на Intel(R) Xeon(R) CPU X5570 @ 2,93 GHz дает на выходе с одного ядра до 70 000 сообщений в секунду и линейно масштабируется при использовании большего числа ядер. Результаты были подтверждены при распределении генерирующих потоков по 8 ядрам и использовании промышленной биржевой системы в качестве мишени. Указанный объем нагрузки, создаваемой с одного сервера, превышает пропускную способность существующих систем торговли акциями. Показатель 70 000 исходящих сообщений в секунду с одного ядра соответствует максимальным показателям инструментов для тестирования веб-инфраструктур с помощью статических запросов [17].

4.6 Настройка профиля нагрузки

В этой части статьи описывается настройка профиля нагрузки. Нагрузка задается параметром:

LOAD_CONFIG = фаза1 [, фаза2, . . . , фазаN]

Нагрузочная фаза может быть следующей:

- const(freq, dur) — постоянная нагрузка с частотой freq и длительностью dur. Возможно также использовать сокращенный формат — freq:dur;
- step(freq, delta, steps, dur) — увеличивающаяся нагрузка с начальной частотой freq, шагом изменения частоты delta, количеством шагов steps и длительностью одного шага dur;
- connect(dur) — все сессии должны установить соединение с задержкой dur;
- disconnect(dur) — все сессии должны оборвать соединение с задержкой dur;
- logon(dur) — все сессии должны послать сообщение с авторизацией с задержкой dur;
- logout(dur) — все сессии должны послать сообщение о прекращении сессии с задержкой dur;

Обрыв соединения сам по себе не является критической проблемой. Например, все веб-соединения, и особенно соединения в мобильных приложениях, рассчитаны на то, что они будут прерываться. Для финансовых протоколов это событие может означать потерю участником торговли контроля над его заявками, и многие системы настроены на отмену всех открытых заявок. Если клиент активно ведет торговлю и выставляет много заявок, потеря соединения приведет к тому, что его заявки будут массово отменены системой, что вызовет повышенную нагрузку на ядро системы. Также необходимо знать, как поведет себя

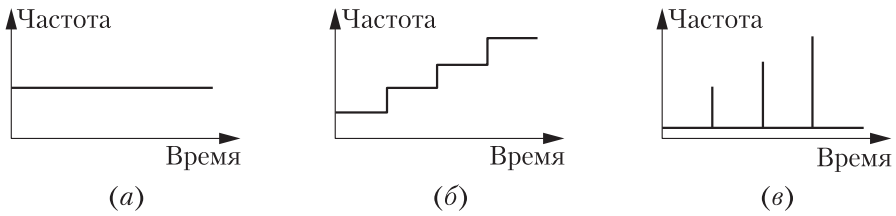


Рис. 2 Простейшие варианты профилей нагрузки: (*а*) const; (*б*) step; (*в*) microburst

система при восстановлении соединения и авторизации под нагрузкой. Очень важно иметь возможность воспроизводить внезапный обрыв и восстановление соединения под нагрузкой. Для этой цели были созданы вышеописанные фазы: connect, disconnect, logon, logout.

На рис. 2 изображены различные нагрузочные профили const, step и microburst:
const: LOAD_CONFIG = const(1000, 20m);
step: LOAD_CONFIG = step(500, 500, 4, 4m);
microburst: LOAD_CONFIG = 200:5m, 40 000:10ms, 200:5m, 75 000:10ms, 200:5m, 100 000:10 ms, 200:5m.

Последний профиль создается при помощи фазы const с малой длительностью и высокой нагрузкой.

Из опыта авторов, нагрузка в форме ступеней (step) в наибольшей степени подходит для определения максимальной производительности системы. Нагрузка в форме микровсплекса в наибольшей степени воспроизводит поведение современных высоконагруженных трейдинговых систем.

5 Заключение

Описанный в статье инструмент успешно используется в рамках проектов, осуществляемых при поддержке компании Exactpro Systems LLC. Достигнутые результаты подтверждают эффективность выбранных методов работы: управления исполняемыми потоками посредством центрального контроллера и использования заготовленных шаблонов при генерации потока сообщений.

Планируется дальнейшее расширение списка открытых и коммерческих протоколов коммуникаций, поддерживаемых данным инструментом для тестирования. Несмотря на то что существующая производительность генератора нагрузки позволяет создать реалистичный поток данных, используя один сервер, достаточный для перегрузки любой из существующих торговых площадок, а также для обеспечения качества трейдиговых систем, которые появятся в ближайшие годы, планируется разработка масштабируемого модуля, позволяющего контролировать нагрузку, создаваемую с нескольких серверов.

Основным направлением исследовательской работы станет совершенствование механизмов обработки обратного потока данных для повышения сложности и реалистичности сценариев нагрузочного тестирования торговых платформ. При этом высокая экономичность и эффективность используемых для тестирования инструментов сохранится.

Литература

1. Foresight: The future of computer trading in financial markets: Final Project Report. — London: The Government Office for Science, 2012. http://www.bis.gov.uk/assets/foresight/docs/computer-trading/12-1086-future-of-computer-trading-in- financial-markets-report.pdf.
2. Roundtable on technology and trading: Promoting stability in today's markets. — Washington, D.C., USA: U.S. Securities and Exchange Commission, October 2, 2012. http://www.sec.gov/news/otherwebcasts/2012/ttr100212-transcript.pdf.
3. Apache: JMeter distributed testing step-by-step. http://jmeter.apache.org/usermanual/jmeter_distributed_testing_step_by_step.pdf.
4. HP LoadRunner for the Windows operating system (Software Ver. 11.50): User Guide. Document Release Date: June 2012. ftp://ftp.itrc.hp.com/applications/HPSoftware/ONLINE_HELP/LoadRunner11.50_User.pdf.
5. IBM: Rational performance tester. http://www-03.ibm.com/software/products/ru/ru/performance.
6. Silk performer: Performance testing tool — test performance under simulated peak stress and load. http://www.borland.com/products/silkperformer.
7. Penhaligan P. Equity trading: Performance, latency & throughput // ExTENT Conference, May 15, 2012. http://www.slideshare.net/extentconf/extent3-turquoise-equitytrading2012.
8. *Benedetti E., Zanetti L.* The focus beyond low latency // ExTENT Conference, March 3, 2013. http://www.slideshare.net/extentconf/extent-2013-obninsk-lse-the-focus-beyond-low-latency.
9. *Cong J.* Load specification and load generation for multimedia traffic loads in computer networks. — Hamburg: FB Informatik, University of Hamburg, 2006. Ph.D. Dissertation.
10. *Cong J., Wolfinger B. E.* A unified load generator based on formal load specification and load transformation // Valuetools '06: 1st Conference (International) on Performance Evaluation Methodologies and Tools Proceedings. — New York, NY, USA: ACM Press, 2006. P. 55–63.
11. *Bodik P., Fox A., Franklin M., Jordan M., Patterson D.* Characterizing, modeling, and generating workload spikes for stateful services // SoCC'10: 1st ACM Symposium on Cloud Computing Proceedings. — New York, NY, USA: ACM, 2010. P. 241–252.
12. *Mosberger D., Jin T.* httperf — a tool for measuring web server performance // SIGMETRICS Performance Eval. Rev., 1998. Vol. 26. No. 3. P. 31–37.
13. *Иткин И Л.* Тестирование биржевых систем в условиях высокочастотного трейдинга // SQA Days-10: Доклады конф., 2014. http://sqadays.com/talk.sdf/sqadays/11151/talks/12196.

14. Yandex.Tank Documentation: Release 1.1.1. — Yandex, March 13, 2014. https://media.readthedocs.org/pdf/yandextank/latest/yandextank.pdf.
15. *Itkin I.* Theory of high frequency trading systems testing // Software Development & Analysis Technologiesin Auditorium Seminar in Lomonosov Moscow State University, 2011. http://sdat.ispras.ru/2011/09/20-октября-модели-тестирования-систем; http://www.slideshare.net/losifItkin/theory-of-high-frequency-trading-systems-testing.
16. Fix Trading Community: Shaping the future of trading. http://www.fixprotocol.org.
17. How to generate millions of HTTP requests. http://dak1n1.com/blog/14-http-load-generate.

Поступила в редакцию 7.02.14

HIGH-PERFORMANCE LOAD GENERATOR FOR HIGH-FREQUENCY TRADING SYSTEMS VERIFICATION

***D. K. Guriev*[1], *M. A. Gai*[2], *I. L. Itkin*[3], *and A. A. Terentiev*[1]**

[1]"ITS-EXPERT," LLC, 38/114 Vavilov Str., Saratov 410012, Russian Federation
[2]"ITS-EXPERT," LLC, 20A/4 Yuzhnoportovy Proyezd 2nd, Moscow 115088, Russian Federation
[3]"Exactpro Systems," LLC, 4040 Civic Center Drive, Suite 200, San Rafael, CA, USA

Abstract: The growing volume of orders generated by HFT (high-frequency trading) systems has posed the challenge of conducting exchange and brokerage systems testing in production-like environments. Specialized testing tools are used to ensure quality of high-load trading systems with high availability. The main requirement for such tools is that they should be capable of creating realistic, high loads using limited hardware infrastructure. The article describes a load injection tool developed for testing automated trading systems and an approach that ensures high performance. The tool is used when throughput and response times of large-scale exchange and brokerage platforms, the backbone of the technological infrastructure of financial markets, are measured.

Keywords: test automation; trading systems; HiVAT

DOI: 10.14357/08696527140205

References

1. The Government Office for Science. 2012. Foresight: The future of computer trading in financial markets: Final Project Report. London. Available at: http://www.bis.gov.uk/assets/foresight/docs/computer-trading/12-1086-future-of-computer-trading-in- financial-markets-report.pdf (accessed March 17, 2014).

2. U.S. Securities and Exchange Commission. 2012. *Roundtable on technology and trading: Promoting stability in today's markets*. Washington, D.C., USA. Available at: http://www.sec.gov/news/otherwebcasts/2012/ttr100212-transcript.pdf (accessed March 17, 2014).
3. Apache: JMeter distributed testing step-by-step. Available at: http://jmeter.apache.org/usermanual/jmeter_distributed_testing_step_by_step.pdf (accessed March 17, 2014).
4. HP LoadRunner for the Windows operating system (Software Version: 11.50): User Guide. Document Release Date: June 2012. Available at: ftp://ftp.itrc.hp.com/applications/HPSoftware/ONLINE_HELP/LoadRunner11.50_User.pdf (accessed March 17, 2014).
5. IBM: Rational Performance Tester. Available at: http://www-03.ibm.com/software/products/ ru/ru/performance (accessed March 17, 2014).
6. Silk performer: Performance testing tool — test performance under simulated peak stress and load. Available at: http://www.borland.com/products/silkperformer (accessed March 17, 2014).
7. Penhaligan, P. 2012. Equity trading: Performance, latency & throughput. *ExTENT Conference*. Available at: http://www.slideshare.net/extentconf/extent3-turquoise-equitytrading2012 (accessed March 17, 2014).
8. *Benedetti, E., and L. Zanetti.* 2013. The focus beyond low latency. *ExTENT Conference*. Available at: http://www.slideshare.net/extentconf/extent-2013-obninsk-lse-the-focus-beyond-low-latency (accessed March 17, 2014).
9. Cong, J. 2006. Load specification and load generation for multimedia traffic loads in computer networks. Hamburg: FB Informatik, University of Hamburg. Ph.D. Dissertation.
10. Cong, J., and B. E. Wolfinger. 2006. A unified load generator based on formal load specification and load transformation. *Valuetools'06: 1st Conference (International) on Performance Evaluation Methodologies and Tools Proceedings*. New York, NY, USA: ACM Press. 55–63.
11. Bodik, P., A. Fox, M. Franklin, M. Jordan, and D. Patterson. 2010. Characterizing, modeling, and generating workload spikes for stateful services. *SoCC'10: 1st ACM Symposium on Cloud Computing Proceedings*. New York, NY, USA: ACM. 241–252.
12. Mosberger, D., and T. Jin. 1998. httperf — a tool for measuring web server performance. *SIGMETRICS Performance Eval. Rev*. 26(3):31–37.
13. Itkin, I. L. 2011. Testirovanie birzhevykh sistem v usloviyakh vysokochastotnogo treydinga [Testing exchange systems under conditions of high frequency trading]. *Conference SQA Days-10 Proceedings*. Moscow. Available at: http://sqadays.com/talk.sdf/sqadays/11151/talks/12196 (accessed March 18, 2014).
14. Yandex. 2014. Yandex.Tank Documentation: Release 1.1.1. Available at: https://media.readthedocs.org/pdf/yandextank/latest/yandextank.pdf (accessed March 17, 2014).
15. Itkin, I. 2011. Theory of high frequency trading systems testing. *Software Development & Analysis Technologiesin Auditorium Seminar in Lomonosov Moscow State University*. Available at: http://sdat.ispras.ru/2011/09/20-октября-модели-тестирования-систем; http://www.slideshare.net/losifltkin/theory-of-high-frequency-trading-systems-testing (accessed March 17, 2014).

16. Fix Trading Community: Shaping the future of trading. Available at: http://www.fixprotocol.org (accessed March 17, 2014).
17. How to generate millions of HTTP requests. Available at: http://dak1n1.com/blog/14-http-load-generate (accessed March 17, 2014).

Received February 07, 2014

Contributors

Guriev Dmitry K. (b. 1978) — senior programmer, "ITS-EXPERT," LLC, 38/114 Vavilov Str., Saratov 410012, Russian Federation; Dmitry.Guriev@exactprosystems.com

Gai Maria A. (b. 1983) — senior analyst ,"ITS-EXPERT," LLC, 20A/4 Yuzhnoportovy Proyezd 2nd, Moscow 115088, Russian Federation; Maria.Gai@exactprosystems.com

Itkin Iosif L. (b. 1978) — Managing Director, "Exactpro Systems," LLC, 4040 Civic Center Drive, Suite 200, San Rafael, CA, USA; Iosif.Itkin@exactprosystems.com

Terentiev Alexander A. (b. 1951) — Doctor of Science in technology, Director of Research, "ITS-EXPERT," LLC, 38/114 Vavilov Str., Saratov 410012, Russian Federation; a_a_terentyev@mail.ru

ИСПОЛЬЗОВАНИЕ ИНСТРУМЕНТОВ ДЛЯ ПАССИВНОГО ТЕСТИРОВАНИЯ ПРИ СЕРТИФИКАЦИИ КЛИЕНТОВ ТРЕЙДИНГОВЫХ СИСТЕМ*

***А. Н. Алексеенко*[1], *А. А. Аверина*[2], *Д. С. Шаров*[3], *П. А. Проценко*[4], *И. Л. Иткин*[5]**

Аннотация: Жизненный цикл разработки биржевого и брокерского программного обеспечения помимо проверки функциональных и технических характеристик системы включает в себя обязательный этап интеграционного тестирования, называемый сертификацией клиентов. Этот этап призван обеспечить совместимость систем автоматизированной торговли, подключаемых к бирже или брокеру посредством финансовых протоколов (таких как Financial Information eXchange (FIX)/FAST, ITCH или специализированные бинарные интерфейсы доступа). В статье представлен оригинальный инструмент, разработанный для проверки совместимости торговых систем. Отличительная особенность разработанного инструмента — унифицированный способ поддержки множества протоколов. Приведены примеры его использования при самостоятельной сертификации участников торгов, а также при масштабных миграциях трейдинговых платформ.

Ключевые слова: финансовые протоколы; FIX-протокол; тестирование совместимости; самостоятельная сертификация; торговый брокер; биржа

DOI: 10.14357/08696527140206

1 Введение

Адаптация новых клиентов к трейдинговой платформе и их поддержка при обновлениях программного обеспечения — это один из ключевых бизнес-процессов биржевых и брокерских организаций [1–3]. Существенной составляющей этого процесса служит тестирование, проводимое для выявления проблем с совместимостью трейдинговой платформы с системами, принадлежащими подключаемым к бирже или брокеру участникам торгов, называемое сертификацией клиентов [4].

*Статья рекомендована к публикации в журнале Программным комитетом конференции «Tools & Methods of Program Analysis» («Инструменты и методы анализа программ», TMPA-2013, 10–12 октября 2013 г., г. Кострома, РФ).

[1]ООО «ИТС-ЭКСПЕРТ», г. Москва, Andrey.Alexeenko@exactprosystems.com

[2]ООО «ИТС-ЭКСПЕРТ», г. Кострома, Nastasya-89@bk.ru

[3]ООО «ИТС-ЭКСПЕРТ», г. Кострома, Daniel.Sharov@exactprosystems.com

[4]Exactpro Systems LLC, США, Калифорния, Pavel.Protsenko@exactprosystems.com

[5]Exactpro Systems LLC, США, Калифорния, Iosif.Itkin@exactprosystems.com

Сертификация клиентов обязательна для любого оператора биржевой или брокерской платформы, предоставляющей возможность осуществлять финансовые транзакции, и в настоящее время широко используется в финансовой индустрии [5]. Ссылки на правила сертификации клиентов и соответствующую документацию можно найти на официальных сайтах соответствующих организаций [6–9].

Программное обеспечение, ранее успешно прошедшее внутреннее интеграционное тестирование, проходит после этого через завершающий тест — интеграцию с клиентами (обычно юридическими лицами). Общепринятой практикой является ситуация, когда клиенты и оператор трейдинговой платформы задают временной интервал для выполнения активного тестирования и оценивают его результаты с обеих сторон.

В связи с тем, что в процесс сертификационного тестирования вовлечены люди, представляющие разные компании, данный процесс требует значительной степени координации и слаженности работы. Это чрезвычайно важно как с финансовой, так и с репутационной точки зрения. Любой дефект программного обеспечения, обнаруженный на этапе сертификационного тестирования, является достаточно дорогостоящим с точки зрения его устранения, поскольку все стадии жизненного цикла программного обеспечения уже пройдены [10], а любая задержка с обнаружением проблем совместимости вносит существенные дополнительные затраты.

К характерным проблемам, с которыми приходится сталкиваться при проведении сертификации клиентов, относятся:

- наличие часовых поясов и возникающие в связи с этим сложности в планировании и координации проведения тестирования командами профессионалов, находящимися физически в разных частях земного шара;
- производительность: сертификация может потребоваться нескольким клиентам одновременно из-за бизнес- или технических событий, таких как выход новых версий программного обеспечения, изменения нормативных или технических требований;
- компетентность: специалист по обеспечению качества программного обеспечения, проводящий сертификацию, обязан обладать должным уровнем технических и бизнес-знаний;
- покрытие: сертификационные тесты должны быть основаны на типичных сценариях и обеспечивать значимые результаты.

В трейдинговой индустрии начинает формироваться понимание того, что одним из способов преодоления этих проблем может стать создание более совершенных решений по автоматизации сертификационного тестирования и анализу его результатов [11]. Несмотря на актуальность автоматизации процесса сертификации биржевых/брокерских клиентов, данная проблематика пока практически не освещена в отечественной и зарубежной научной литературе.

На рынке присутствуют несколько коммерческих инструментов тестирования, созданных для ускорения процесса сертификации клиентов:

- FIX Conductor от компании Lasalletech [12];
- FACTS (FIX Automated Certification and Testing Service) от B2BITS, EPAM Systems [13];
- CertiFIX от Greenline [14];
- Catalys Studio от Cameron [15];
- Ignition от FIXFlyer [16];
- VegaFABS от Pravega Financial Technologies [17];
- Certpoint от Tradepoint Systems [18];
- Certification Platform от FixSpec.com [19].

Биржевые платформы используют два основных подхода к сертификации:

(1) предоставление клиентам специализированного симулятора для проведения сертификации [20–22]. Клиенты проводят тестирование, используя предоставленный инструмент, и предоставляют логи сертификации;

(2) предоставление доступа к специализированному тестовому окружению, которое является уменьшенной копией промышленной системы; документирование процедуры; совместное проведение сертификации с клиентом либо использование самостоятельной сертификации [23]; использование пассивных методов, таких как перехват трафика сетевых подключений или данные из лог-файлов, для анализа успешности и полноты проведенной процедуры сертификации.

Оба подхода в той или иной степени реализованы в существующих коммерческих решениях.

Основное достоинство метода, основанного на использовании симуляторов, — это снижение нагрузки на сертифицирующую организацию и клиентов, обеспечиваемое возможностью проводить сертификацию в любое время суток, а не только в период открытого торгового дня. Проблема, специфичная для всех симуляторов, — отсутствие гарантии, что фактическое сообщение, отправленное с его помощью, идентично сообщениям, отправляемым реальным программным обеспечением [24].

Пассивное тестирование является, на взгляд авторов, наиболее корректным методом проведения сертификации клиентов. Общая концепция пассивного тестирования и ее преимущества описаны в [25], различные алгоритмы пассивного тестирования предложены в [26, 27]. Основным преимуществом пассивного тестирования является то, что инструмент для тестирования не оказывает влияния на тестируемую систему и не приводит к созданию потоков дополнительных сообщений. Данное преимущество, однако, одновременно является и недостатком: для успешного проведения пассивного тестирования необходимые

сообщения должны быть кем-то созданы. В случае сертификации клиентов требуется привлекать представителей другой компании для создания входящего потока сообщений.

Данная практика неизбежна при первичной сертификации. Однако с целью снижения временны́х затрат сторонних организаций требуется использовать более эффективные методы: например, предложенный в [28] подход. Используя данные первичной сертификации для автоматизированного создания новых сценариев тестирования при сохранении неизменной спецификации протокола доступа, оператор трейдинговой платформы получает возможность провести повторную сертификацию клиента самостоятельно.

Присутствующие на рынке инструменты для сертификации клиентов обладают следующими ограничениями:

– специализация на одном фиксированном протоколе, например FIX [29];

– интерактивные инструменты анализа данных и создания новых сценариев малопригодны для обработки действительно больших объемов данных о гетерогенных клиентских подключениях.

Авторы принимают участие в разработке и использовании программного решения компании Exactpro Systems LLC, направленного на преодоление указанных ограничений [30, 31]. Созданный инструмент описывается во второй части данной статьи. Третья часть содержит описание практического использования созданного инструмента для самостоятельной сертификации участников биржевых торгов. Четвертая часть описывает использование разработанного инструмента при масштабных миграциях брокерских систем.

2 Многопротокольное решение для пассивного тестирования трейдинговых систем

Инструмент для пассивного тестирования сетевых подключений к трейдинговым системам разработан с использованием языка Java и системы управления базами данных MySQL. В основе подхода — унифицированное описание структуры протокольных сообщений: система словарей. Для каждого протокола создается словарь. Для групп протоколов может потребоваться написание специальных модулей, приводящих сетевые потоки данных к унифицированному внутреннему формату на основе словарей. Концепция близка к логике Complex Events Processing [32]. Для создания шаблона словаря был взят и доработан для большей общности шаблон словарей, используемый в системе QuickFIX/J [33].

Разработанный инструмент может получать входящие данные о перехваченных сетевых сообщениях, полученных с использованием tcpdump [34] или различных прокси-серверов [35]. Пользователю также предоставлена возможность загружать лог-файлы, содержащие массив сообщений, в настраиваемом формате.

Основная таблица в базе данных содержит информацию о каждом перехваченном пакете, включая повторную пересылку TCP (Transmission Control Protocol) данных. Если трейдинговая система находится под нагрузкой, сетевой пакет может содержать несколько сообщений или же сообщение может быть разбито между несколькими TCP-пакетами. Выделив сообщения из сетевых

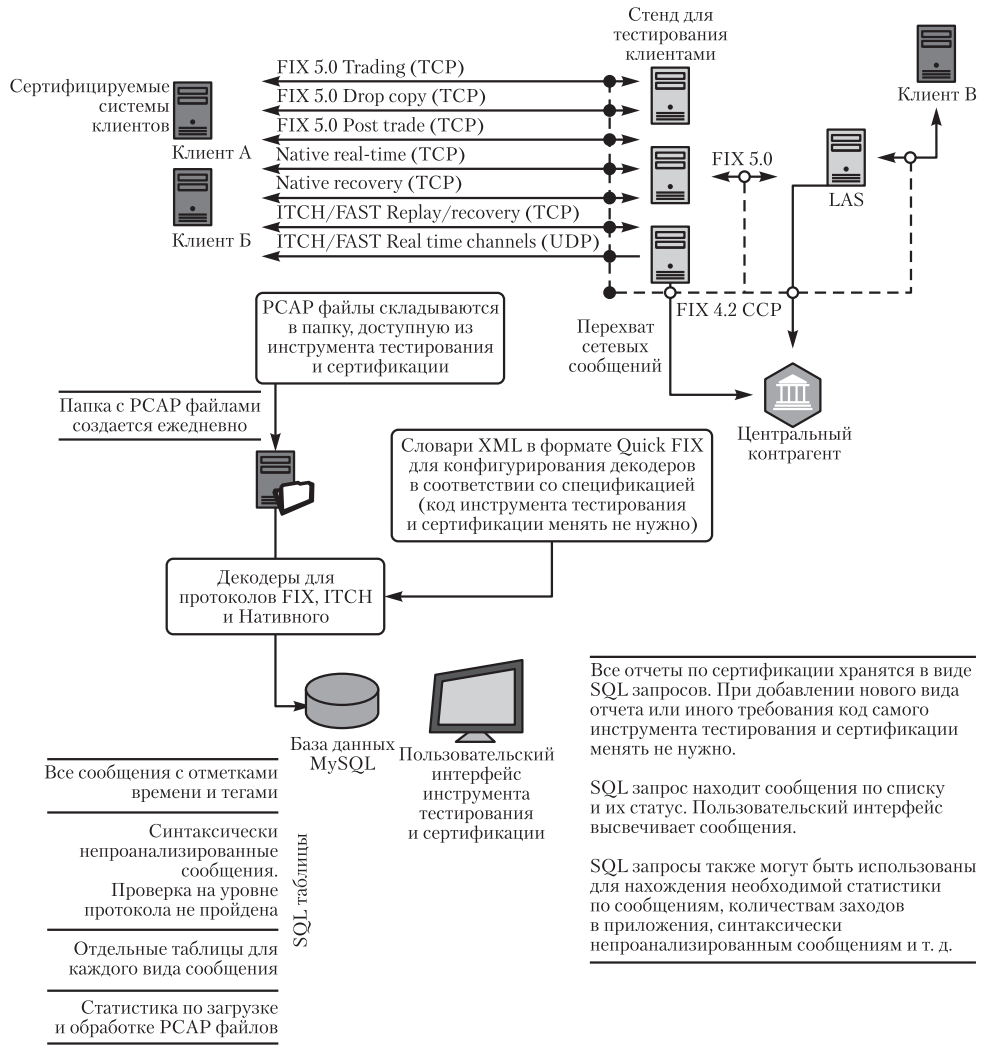


Рис. 1 Процесс обработки сообщений в инструменте тестирования и сертификации (UDP — User Datagram Protocol; CCP — Compression Control Protocol; PCAP — packet capture; LAS — Live Access Server; XML — eXtensible Markup Language

пакетов, инструмент делает попытку конвертации в унифицированный формат на основе словарей. Информация о сетевых пакетах и сообщениях, не прошедших проверку посредством словарей, заносится в отдельную таблицу, по сути содержащую список проблем сертификации первого уровня.

Детали перехваченных и успешно обработанных сообщений могут быть сохранены в реляционную базу данных посредством двух основных методов:

(1) использования общей таблицы, содержащей идентификаторы сообщения, имя параметра и его значение;

(2) использования индивидуальной таблицы для каждого типа сообщения, размещая параметры сообщения в соответствующих колонках с именами, произведенными от имен параметров.

Преимущество первого подхода — это большая общность. Преимущество второго подхода — удобство работы с SQL (Structured Query Language) запросами к базе данных. В разработанном инструменте используется комбинация обоих подходов: для каждого типа сообщений создана индивидуальная таблица, а данные из повторяющихся групп хранятся в общей таблице. Рисунок 1 содержит схему использования разработанного инструмента для перехвата и хранения сетевых сообщений.

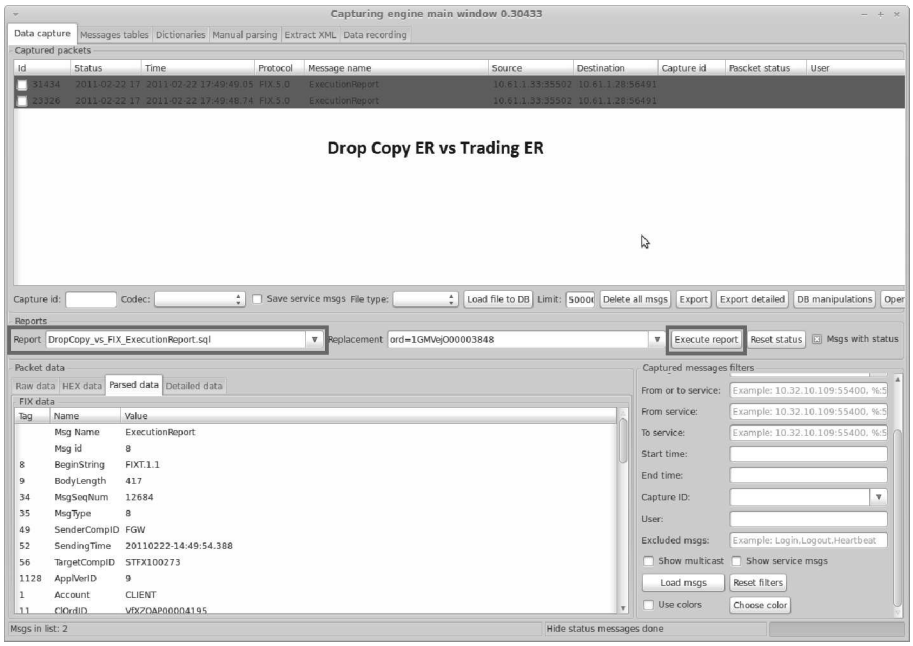


Рис. 2 Графический пользовательский интерфейс инструмента тестирования и сертификации

Перехваченные сообщения декодируются и складываются в базу данных инструмента. Все отчеты по сертификации хранятся в виде SQL-запросов. При добавлении нового вида отчетов или иного требования нет необходимости менять код самого инструмента.

С помощью SQL-запроса можно найти сообщения по списку и их статус и выделить их цветом с помощью пользовательского интерфейса для лучшего зрительного восприятия. SQL-запросы также могут быть использованы для нахождения необходимой статистики по сообщениям, числу заходов в приложения, синтаксически не проанализированным сообщениям и т. д.

Графический интерфейс пользователя разработанного инструмента позволяет аналитику, отвечающему за сертификацию или обеспечение качества трейдинговой платформы, просматривать в реальном режиме времени сообщения и события, включая детали сетевых пакетов, результаты верификации посредством системы словарей, значения индивидуальных полей и исходные бинарные данные.

Сертификационные тесты и сверки данных могут быть реализованы посредством обычных SQL-запросов. Инструмент предоставляет возможность подсветки и анализа сообщений, полученных в результате выполнения того или иного запроса. На рис. 2 представлен пример окна графического интерфейса разработанного инструмента.

3 Самостоятельная сертификация участников биржевых торгов

Регулирующие органы требуют от операторов биржевой системы предоставления эквивалентного доступа всем участникам торгов [36]. При внесении изменений в технологическую платформу биржи ее оператор обязан провести сертификацию всех подключенных торговых систем в короткий промежуток времени. Данная особенность процесса создает существенную нагрузку на отделы организации — оператора биржевой площадки, отвечающие за поддержку клиентов. Для снижения этой нагрузки биржевые площадки используют метод самостоятельной сертификации.

Клиентам предоставляется доступ к тестовому окружению и сценарий выполнения сертификационных тестов. После выполнения шагов предоставленного сценария клиент высылает логи процесса в сертифицирующую организацию, где они проходят дополнительную проверку.

Пассивное тестирование значительно упрощает процесс сертификации для участников торгов. При данном подходе от них требуется только подключиться к тестовому окружению и отправить заявки, обозначенные в тестовом сценарии. Таким образом, участники торгов избавлены от выполнения любых шагов (таких как установка дополнительного программного обеспечения, сбор логов и т. д.), кроме тех, которые непосредственно необходимы для подключения к технологической платформе биржи.

Разработанный инструмент перехватывает все сообщения, переданные от клиента к бирже или в обратную сторону, разбирает структуру и содержимое

каждого сообщения и сохраняет все в реляционную базу данных. Используя SQL-запросы, аналитик сертифицирующей организации может получить статистику по попыткам выполнения шагов тестового сценария и их успешности для каждого из участников торгов. Данные о сетевых пакетах позволяют также обнаружить дополнительные проблемы, такие как разрывы соединений или проблемы с буферизацией сообщений.

Приведем пример сертификационного SQL-сценария, проверяющего успешность размещения клиентом агрессивной рыночной заявки, в результате появления которой по исполнении ее против заявок, размещенных на противоположной стороне стакана заявок, образовались две и более сделки:

```
insert into t_native_testcases
(user,sourceip,sourceport,testcase,timestamp,clordid,ord
erid,otherid)
select distinct n.user, n.sourceip, n.sourceport, 'MEx-
012.2 Agg. MO' as testcase, n.timestamp, n.clordid,
e.orderid, ''
from t_lsenative_neworder n
, t_lsenative_executionreport e
, t_lsenative_executionreport e2
where n.user=e.user
and n.sourceip=e.destinationip and
n.sourceport=e.destinationport
and n.clordid=e.clordid and n.user=e2.user and
n.sourceip=e2.destinationip
and n.sourceport=e2.destinationport and
n.clordid=e2.clordid
and n.ordertype=1 and e.ordstatus=1 and
e.tradeliquidityindicator='R'
and e2.typeoftrade='2' and e2.ordstatus in (1,2)
and e2.tradeliquidityindicator='R' and
e2.typeoftrade='2'
and e.execid <> e2.execid
order by user, clordid, orderid
```

Для нескольких биржевых платформ были разработаны совокупности SQL-сценариев, покрывающие все требования по сертификации клиентских соединений, опубликованные организатором торгов [8].

4 Миграция брокерской платформы с большим числом гетерогенных клиентских подключений

В сравнении с биржевыми платформами брокерские системы обычно предоставляют существенно более широкие возможности по настройке протокола взаимодействия с системами клиентов: например, используя возможности таких

систем, как UL Bridge [37]. Один и тот же клиент может использовать одновременно несколько брокеров для получения доступа на биржу. Часто брокеры стараются снизить количество изменений в протоколах коммуникации, которые конкретный клиент вынужден имплементировать со своей стороны. Облегчая взаимодействие с клиентом, этот подход одновременно приводит к появлению большого количества гетерогенных конфигураций со стороны брокерской трейдинговой платформы. При внутренних изменениях брокерской платформы возникает необходимость в регрессионном тесте на совместимость с клиентскими системами.

Разработанный инструмент позволяет обработать имеющиеся данные из тестовых и промышленных окружений и создать необходимый набор активных тестовых сценариев. Выполнив эти сценарии против тестового окружения без привлечения конечных клиентов, тестовый инструмент запускает SQL-сценарии, выполняющие сверку ответов брокерской платформы до и после изменений. Особенность созданного авторами инструмента состоит в том, что он позволяет проводить этот процесс одновременно для большого числа клиентов, подключений, рынков и типов сообщений. Гибкость при создании сверяющих SQL-запросов позволяет не только исключить из сравнения поля, про которые заранее известно, что их значения должны различаться (например, sequence numbers, timestamps и др.), но и задать требования по обработке более сложных расхождений.

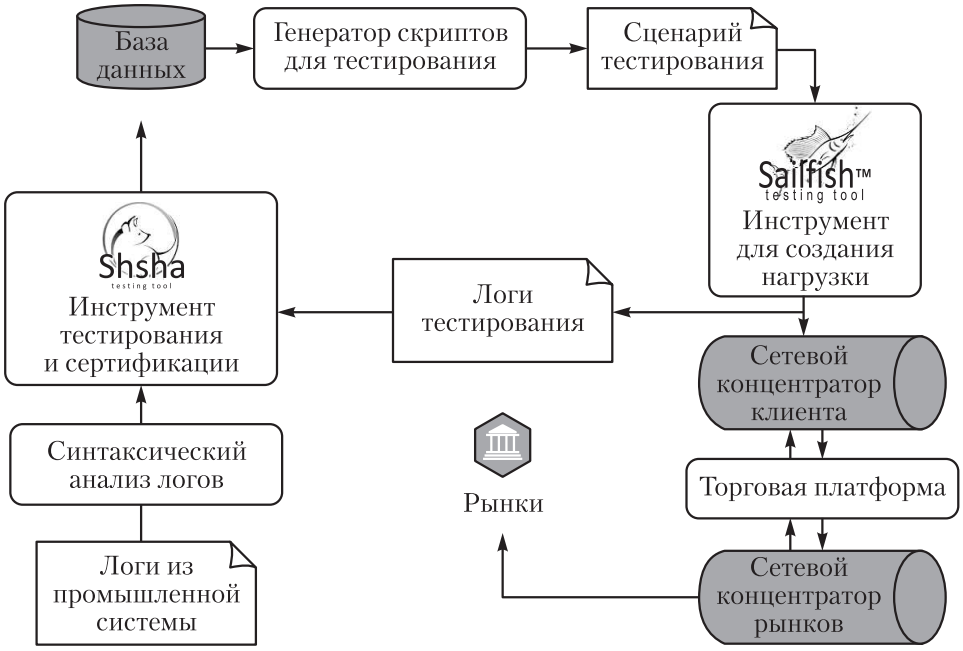


Рис. 3 Схематичный пример реализации предлагаемого решения

Подход доказал свою жизнеспособность и эффективность при миграции трейдинговой платформы одного из крупнейших международных брокеров, предоставляющего доступ к торговле производными финансовыми инструментами. Последовательность шагов отображена на схеме (рис. 3).

Логи из промышленной системы загружаются в базу данных инструмента для тестирования и сертификации. Генератор скриптов использует данную базу для создания совокупности тестовых сценариев, исполняемых активным инструментом для функционального тестирования. Логи выполнения тестовых сценариев загружаются в базу данных для сравнения с данными, полученными из промышленной системы.

5 Заключение

С ростом объемов автоматизированной электронной торговли стабильность и устойчивость финансовых рынков будут во все большей степени зависеть от корректной совместной работы платформ, обеспечивающих инфраструктуру рынков, и подключенных к ним автоматизированных трейдинговых систем. Стоимость процессов подключения новых клиентов, их сертификации и сохранения совместимости при регулярно вносимых изменениях существенно влияет на общую стоимость функционирования биржевых и брокерских систем.

Представленный в статье инструмент используется с целью повышения эффективности и экономичности указанных процессов. Имеется позитивный опыт его применения в рамках проектов компании Exactpro Systems LLC на заключительных этапах выпуска в промышленную эксплуатацию крупномасштабных трейдинговых систем для торговли финансовыми инструментами разных классов.

В статье описаны используемые в индустрии методы сертификации клиентских подключений и сделаны выводы о преимуществах подхода, основанного на использовании многопротокольного пассивного инструмента, сохраняющего данные в реляционную систему управления базами данных. Рассмотрен также вариант использования данного инструмента с целью создания активных тестов и сокращения затрат при масштабных миграциях систем с множеством гетерогенных клиентских подключений.

Дальнейшие исследования будут направлены на оптимизацию структуры базы данных: в частности, в вопросе распространения разработанных методов на трейдинговые протоколы, основанные не на сетевых взаимодействиях, а на программных интерфейсах доступа (API).

Литература

1. Shock of the new // Markit Magazine, Winter 2010. Iss. 10 — Focus: Client Onboarding. http://www.markit.com/assets/en/docs/markit-magazine/issue-10/mm10_focus5-onboardingroundtable.pdf.

2. *Pierron A., Jaswal A.* Institutional client on-boarding in the financial industry: Time to move to the industrialization phase. Celent Report, 2012. http://www.celent.com/reports/institutional-client-boarding-financial-industry.
3. Client onboarding: Solving challenges, maximizing opportunities // Fenergo, April 2013. http://www.fenergo.com/industry-knowledge/whitepapers/client-onboarding-solving-challenges-maximizing-opportunities.html.
4. Challenges and solutions to onboarding trading clients. http://www.trdpnt.com/challenges-solutions-onboarding-trading-clients.
5. FIA market access risk management recommendations, 2010. http://www. futuresindustry.org/downloads/Market_Access-6.pdf.
6. BOX Options Exchange. Software Certification. http://boxexchange.com/what-you-need-to-know/software-certification.
7. Eris exchange electronic trading platform certification process. Eris Exchange, LLC, 2013. http://www.erisfutures.com/sites/default/files/Electronic_Trading_Certification_Process.pdf.
8. TQ 601 — Technical specification: Turquoise equities guide to certification, 2013. Iss. 2.2. http://www.lseg.com/sites/default/files/content/documents/TQ601%20-%20Guide%20To%20Application%20Certifcation.pdf.
9. Порядок сертификации ВПТС ОАО Московская биржа // Московская биржа, 2013. http://fs.moex.com/files/4531.
10. *Shull F., Basili V., Boehm B., Brown A. W., Costa P., Lindvall M., Port D., Rus I., Tesoriero R., Zelkowitz M.* What we have learned about fighting defects // METRICS '02: 8th Symposium (International) on Software Metrics Proceedings. — Washington, DC, USA: IEEE Computer Society, 2002. P. 249–258. http://www.cs.umd.edu/∼basili/publications/proceedings/P95.pdf.
11. *Edelen C.* Making the case for automated FIX certification // Wall Street & Technology, May 24, 2013. http://www.advancedtrading.com/high-frequency/making-the-case-for-automated-fix-certif/240155554.
12. Lasalletech. FIX Conductor^{TM}: Test, certify and onboard — with the ease of automation. http://www.lasalletech.com/products/fix-automated-onboarding.
13. B2BITS, EPAM Systems. FACTS 2.0 — FIX Automated Certification and Testing. http://www.b2bits.com/trading_solutions/fix_testing_facts.html.
14. Greenline Financial Technologies, Inc. CertiFIX®: Automate and simplify your FIX certification process, 2014. http://www.greenlinetech.com/products/certifix.php (18.03.14).
15. Cameron. Catalys studio: Enhance analysis quality and support for your FIX-based business. http://www.camerontecgroup.com/products/pdf/Catalys-Studio-US.pdf.
16. FIXFlyer. Ignition: Certification and trade validation, 2009. http://fixflyer.com/materials/software/Ki/FlyerIgnition.pdf.
17. Pravega® Financial Technologies. VegaFABS, 2009. http://www.pravegatech.com/index.php?option=com_content&view=article&id=64&Itemid=65.
18. Tradepoint Systems. Certpoint: Client onboarding & certification. http://www.trdpnt.com/certpoint.
19. *Siddiqui A.* FixSpec launches innovative multi-venue certification utility to streamline connectivity // Forex Magnates, February 27, 2013. http://forexmagnates.

com/fixspec-launches-innovative-multi-venue-certification-utility-to-streamline-connectivity.

20. HKEx Orion Central Gateway (OCG) & HKEx Orion Market Data Platform (OMD) — simulator packages // The Stock Exchange of Hong Kong Ltd., December 31, 2012. http://www.hkex.com.hk/eng/market/partcir/sehk/2012/Documents/CTMO06612E.pdf.
21. BM&FBOVESPA. MyCTC (Certification and Testing Center): User Guide. Ver. 1.1. Last modified: 24.05.2013. http://www.bmfbovespa.com.br/en-us/services/download/MyCTC-User-Guide.pdf.
22. CME Group. Client System Certification. http://www.cmegroup.com/confluence/display/EPICSANDBOX/Client+System+Certification.
23. Customer Certification and Testing Services // The London Stock Exchange. November 23, 2013. http://www.lseg.com/areas-expertise/technology/market-connectivity/customer-certification-and-testing-services.
24. *Zverev A., Bulda A.* Exchange simulators for SOR/Algo testing: Advantages vs. shortcomings // ExTENT Conference, 2011. http://www.slideshare.net/losifltkin/exchange-simulators-for-sor-algo-testing-advantages-vs-shortcomings.
25. *Brzezinski K. M.* Towards the methodological harmonization of passive testing across ICT communities // Engineering the Computer Science and IT. — InTech, 2009. P. 143–169. http://cdn.intechopen.com/pdfs-wm/8939.pdf.
26. *Lee D., Chen D., Ruibing H., Miller R. E., Jianping Wu, Xia Y.* Network protocol system monitoring — a formal approach with passive testing // IEEE/ACM Trans. Networking (TON), 2006. Vol. 14. No. 2. P. 424–437. http://dl.acm.org/citation.cfm?id=1217634.
27. *Cavalli A., Maag S., Montes de Oca E.* A passive conformance testing approach for a MANET routing protocol // SAC'09: 2009 ACM Symposium on Applied Computing Proceedings. — New York, NY, USA: ACM, 2009. P. 207–211. http://dl.acm.org/citation.cfm?id=1529326.
28. *Newsome J., Brumley D., Franklin J., Song D.* Replayer: Automatic protocol replay by binary analysis // CCS'06: 13th ACM Conference on Computer and Communications Security Proceedings. — Pittsburgh, PA, USA: Carnegie Mellon University, 2006. P. 311–321. http://bitblaze.cs.berkeley.edu/papers/replayer-ccs2006.pdf.
29. FIX Trading Community: Shaping the future of trading. http://www.fixtradingcommunity.org.
30. *Zaitseva N., Popovchuk N.* The next step in reconciliation testing // ExTENT Conference, 2012. http://www.slideshare.net/extentconf/extent3-exactpro-thenextstepinreconciliationtesting.
31. *Zverev A., Moskaleva O., Kudryavtseva M., Doronichev D., Bulda A.* Four houses — test tools presentation // ExTENT Conference, 2012. http://www.slideshare.net/extentconf/extent3-exactpro-fourhousestesttools2012-1.
32. *Cugola G., Margara A.* Processing flows of information: From data stream to complex event processing // ACM Computing Surveys (CSUR), 2012. Vol. 44. Iss. 3. Article No. 15. P. 359–360. http://dl.acm.org/citation.cfm?id=2187677.
33. QuickFIX/J: 100% Java open source FIX engine. http://quickfixj.org.
34. *Fuentes F., Kar D. C.* Ethereal vs. Tcpdump: A comparative study on packet sniffing tools for educational purpose // J. Computing Sci. Colleges, 2005. Vol. 20. Iss. 4. P. 169–176. http://dl.acm.org/citation.cfm?id=1047873.

35. *Aston Ph., Fitzgerald C.* TCP proxy // The Grinder, 26.11.2013. http://grinder.sourceforge.net/g3/tcpproxy.html.
36. Investment Services Directive — Markets in Financial Instruments Directive (MiFID). The EU Single Market. http://ec.europa.eu/internal_market/securities/isd/mifid/index_en.htm.
37. FIX Solutions. Ullink. http://www.ullink.com/fix-solutions.php.

Поступила в редакцию 13.03.14

USAGE OF PASSIVE TESTING TOOLS FOR CERTIFICATION OF TRADING SYSTEMS CLIENTS

***A. N. Alexeenko*[1], *A. A. Averina*[2], *D. S. Sharov*[2], *P. A. Protsenko*[3], *and I. L. Itkin*[3]**

[1]"ITS-EXPERT," LLC, 20A/4 Yuzhnoportovy Proyezd 2nd, Moscow 115088, Russian Federation
[2]"ITS-EXPERT," LLC, 20 Lenin Str., Kostroma 156013, Russian Federation
[3]"Exactpro Systems," LLC, 4040 Civic Center Drive, Suite 200, San Rafael, CA, USA

Abstract: The life cycle of exchange and brokerage software development, along with verification of functional and technical characteristics of the system, includes a mandatory stage of integration testing called clients certification. The stage is designed to ensure the compatibility of automated trading systems which are connected by means of financial protocols (such as FIX/FAST, ITCH, or specialized binary access interfaces) to an exchange or a broker. The article presents a bespoke tool developed to verify the compatibility of trading systems. The distinctive feature of the tool is a unified way of supporting multiple protocols. The article also provides a few examples of using the tool in self-certification of trading participants and during large-scale migrations of trading platforms.

Keywords: financial protocols; FIX-protocol, compatibility testing, self-certification; trading broker; stock exchange

DOI: 10.14357/08696527140206

References

1. Shock of the new. 2010. Focus: Client Onboarding. *Markit Magazine*, Winter 2010. Iss. 10. Available at: http://www.markit.com/assets/en/docs/markit-magazine/issue-10/mm10_focus5-onboardingroundtable.pdf (accessed March 18, 2014).
2. Pierron A., and A. Jaswal. 2012. Institutional client on-boarding in the financial industry: Time to move to the industrialization phase. Celent Report. Available at: http://www.celent.com/reports/institutional-client-boarding-financial-industry (accessed March 18, 2014).

3. Client onboarding: Solving challenges, maximizing opportunities. April 2013. *Fenergo*. Available at: http://www.fenergo.com/industry-knowledge/whitepapers/client-onboarding-solving-challenges-maximizing-opportunities.html (accessed March 18, 2014).
4. Challenges and solutions to onboarding trading clients. Available at: http://www.trdpnt.com/challenges-solutions-onboarding-trading-clients (accessed March 18, 2014).
5. FIA Market access risk management recommendations. April 2010. Available at: http://www.futuresindustry.org/downloads/Market_Access-6.pdf (accessed March 18, 2014).
6. BOX Options Exchange. Software Certification. Available at: http://boxexchange.com/what-you-need-to-know/software-certification (accessed March 18, 2014).
7. Eris Exchange, LLC. 2013. Eris exchange electronic trading platform certification process. Available at: http://www.erisfutures.com/sites/default/files/Electronic_Trading_Certification_Process.pdf (accessed March 18, 2014).
8. TQ 601 — Technical specification: Turquoise equities guide to certification. July 29, 2013. Iss. 2.2. Available at http://www.lseg.com/sites/default/files/content/documents/TQ601%20-%20Guide%20To%20Application%20Certifcation.pdf (accessed March 18, 2014).
9. Moskovskaya Birzha [Moscow Stock Exchange]. 2013. *Poryadok sertifikacii VPTS OAO Moskovskaya birzha* [Procedure for external software and hardware certification of Open Joint Stock Company Moscow Stock Exchange]. Available at: http://fs.moex.com/files/4531 (accessed March 18, 2014).
10. Shull, F., V. Basili, B. Boehm, A. W. Brown, P. Costa, M. Lindvall, D. Port, I. Rus, R. Tesoriero, and M. Zelkowitz. 2002. What we have learned about fighting defects. *METRICS'02: 8th Symposium (International) on Software Metrics Proceedings*. Washington, DC, USA: IEEE Computer Society. 249–258. Available at: http://www.cs.umd.edu/∼basili/publications/proceedings/P95.pdf (accessed March 18, 2014).
11. Edelen, C. May 24, 2013. Making the case for automated FIX certification. *Wall Street & Technology*. Available at: http://www.advancedtrading.com/high-frequency/making-the-case-for-automated-fix-certif/240155554 (accessed March 18, 2014).
12. Lasalletech. FIX Conductor™: Test, certify and onboard — with the ease of automation. Available at: http://www.lasalletech.com/products/fix-automated-onboarding (accessed March 18, 2014).
13. B2BITS, EPAM Systems. FACTS 2.0 — FIX Automated Certification and Testing. Available at: http://www.b2bits.com/trading_solutions/fix_testing_facts.html (accessed March 18, 2014).
14. Greenline Financial Technologies, Inc. 2014. CertiFIX®: Automate and simplify your FIX certification process. Available at: http://www.greenlinetech.com/products/certifix.php (accessed March 18, 2014).
15. Cameron. Catalys studio: Enhance analysis quality and support for your FIX-based business. Available at: http://www.camerontecgroup.com/products/pdf/Catalys-Studio-US.pdf (accessed March 18, 2014).
16. FIXFlyer. 2009. Ignition: Certification and trade validation. Available at: http://fixflyer.com/materials/software/Ki/FlyerIgnition.pdf (accessed March 18, 2014).

17. Pravega® Financial Technologies. 2009. VegaFABS. Available at: http://www.pravegatech.com/index.php?option=com_content&view=article&id=64&Itemid=65 (accessed March 18, 2014).
18. Tradepoint Systems. Certpoint: Client onboarding & certification. Available at: http://www.trdpnt.com/certpoint (accessed March 18, 2014).
19. Siddiqui, A. February 27, 2013. FixSpec launches innovative multi-venue certification utility to streamline connectivity. *Forex Magnates*. Available at: http://forexmagnates. com/fixspec-launches-innovative-multi-venue-certification-utility-to-streamline-connectivity (accessed March 18, 2014).
20. The Stock Exchange of Hong Kong Ltd. 2012. HKEx Orion Central Gateway (OCG) & HKEx Orion Market Data Platform (OMD) — simulator packages. Available at: http://www.hkex.com.hk/eng/market/partcir/sehk/2012/Documents/CTMO06612E.pdf (accessed March 18, 2014).
21. BM&FBOVESPA. MyCTC (Certification and Testing Center): User Guide. Ver. 1.1. Last modified: 24.05.2013. Available at: http://www.bmfbovespa.com.br/en-us/services/download/MyCTC-User-Guide.pdf (accessed March 18, 2014).
22. CME Group. Client System Certification. Available at: http://www.cmegroup.com/confluence/display/EPICSANDBOX/Client+System+Certification (accessed March 18, 2014).
23. Customer Certification and Testing Services. *The London Stock Exchange*. November 23, 2013. Available at: http://www.lseg.com/areas-expertise/technology/market-connectivity/customer-certification-and-testing-services (accessed March 18, 2014).
24. Zverev, A., and A. Bulda. 2011. Exchange simulators for SOR/Algo testing: Advantages vs. shortcomings. *ExTENT Conference*. Available at: http://www.slideshare.net/losifltkin/exchange-simulators-for-sor-algo-testing-advantages-vs-shortcomings (accessed March 18, 2014).
25. Brzezinski, K. M. 2009. Towards the methodological harmonization of passive testing across ICT communities. *Engineering the computer science and IT*. InTech. 143–169. Available at: http://cdn.intechopen.com/pdfs-wm/8939.pdf (accessed March 18, 2014).
26. Lee, D., D. Chen, H. Ruibing, R. E. Miller, Wu Jianping, and Y. Xia. 2006. Network Protocol System Monitoring — a formal approach with passive testing. *IEEE/ACM Trans. Networking (TON)* 14(2):424–437. Available at: http://dl.acm.org/citation.cfm?id=1217634 (accessed March 18, 2014).
27. Cavalli, A., S. Maag, and E. Montes de Oca. 2009. A passive conformance testing approach for a MANET routing protocol. *SAC'09: 2009 ACM Symposium on Applied Computing Proceedings*. New York, NY, USA: ACM. 207–211. Available at: http://dl.acm.org/citation.cfm?id=1529326 (accessed March 18, 2014).
28. Newsome, J., D. Brumley, J. Franklin, and D. Song. 2006. Replayer: Automatic protocol replay by binary analysis. *CCS'06: 13th ACM Conference on Computer and Communications Security Proceedings*. Pittsburgh, PA, USA: Carnegie Mellon University. 311–321. Available at: http://bitblaze.cs.berkeley.edu/papers/replayer-ccs2006.pdf (accessed March 18, 2014).
29. FIX Trading Community: Shaping the future of trading. Available at: http://www.fixtradingcommunity.org (accessed March 18, 2014).

30. Zaitseva, N., and N. Popovchuk. 2012. The next step in reconciliation testing. *ExTENT Conference* Available at: http://www.slideshare.net/extentconf/extent3-exactpro-thenextstepinreconciliationtesting (accessed March 18, 2014).
31. Zverev, A., O. Moskaleva, M. Kudryavtseva, D. Doronichev, and A. Bulda. 2012. Four houses — test tools presentation. *ExTENT Conference*. Available at: http://www.slideshare.net/extentconf/extent3-exactpro-fourhousestesttools2012-1 (accessed March 18, 2014).
32. Cugola, G., and A. Margara. 2012. Processing flows of information: From data stream to complex event processing. *ACM Computing Surveys (CSUR)*. Article No. 15. 44(3):359–360. Available at: http://dl.acm.org/citation.cfm?id=2187677 (accessed March 18, 2014).
33. QuickFIX/J: 100% Java open source FIX engine. Available at: http://quickfixj.org (accessed March 18, 2014).
34. Fuentes, F., and D. C. Kar. 2005. Ethereal vs. Tcpdump: A comparative study on packet sniffing tools for educational purpose. *J. Computing Sci. Colleges* 20(4):169–176. Available at: http://dl.acm.org/citation.cfm?id=1047873 (accessed March 18, 2014).
35. Aston, Ph., and C. Fitzgerald. 2013. TCP proxy. *The Grinder*. Available at: http://grinder. sourceforge.net/g3/tcpproxy.html (accessed March 18, 2014).
36. Investment Services Directive — Markets in Financial Instruments Directive (MiFID). *The EU Single Market*. Available at: http://ec.europa.eu/internal_market/securities/isd/mifid/index_en.htm (accessed March 18, 2014).
37. FIX Solutions. *Ullink*. Available at: http://www.ullink.com/fix-solutions.php (accessed March 18, 2014).

Received March 13, 2014

Contributors

Alexeenko Andrey N. (b. 1983) — Head of Department, "ITS-EXPERT," LLC, 20A/4 Yuzhnoportovy Proyezd 2nd, Moscow 115088, Russian Federation; Andrey.Alexeenko@exactprosystems.com

Averina Anastasia A. (b. 1989) — Head of Department, "ITS-EXPERT," LLC, 20 Lenin Str., Kostroma 156013, Russian Federation; Nastasya-89@bk.ru

Sharov Daniel S. (b. 1991) — programmer, "ITS-EXPERT," LLC, 20 Lenin Str., Kostroma 156013, Russian Federation; Daniel.Sharov@exactprosystems.com

Protsenko Pavel A. (b. 1983) — senior project manager, "Exactpro Systems," LLC, 4040 Civic Center Drive, Suite 200, San Rafael, CA, USA; Pavel.Protsenko@exactprosystems.com

Itkin Iosif L. (b. 1978) — Managing Director, "Exactpro Systems," LLC, 4040 Civic Center Drive, Suite 200, San Rafael, CA, USA; Iosif.Itkin@exactprosystems.com

ТЕХНОЛОГИЯ АНАЛИЗА ИСХОДНОГО КОДА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ЧАСТИЧНЫХ СПЕЦИФИКАЦИЙ ДЛЯ АВТОМАТИЗИРОВАННОЙ ГЕНЕРАЦИИ ТЕСТОВ*

А. А. Андрианова[1], *В. М. Ицыксон*[2]

Аннотация: Повышение качества создаваемого программного обеспечения (ПО) является одной из основных проблем программной инженерии. Одним из путей повышения качества программ является автоматизируемая генерация тестов. В настоящей статье предлагается технология автоматизированного создания модульных тестов, комбинирующая функциональный и структурный подходы. Для обеспечения покрытия тестами путей программы используется информация, извлекаемая из исходного кода программы, а для формирования тестовых оракулов и определения параметров тестов используются частичные спецификации, заданные в форме контрактов. Разработанный подход реализован в виде инструментального прототипа, анализирующего программы на языке Java и формирующего тест-кейсы для методов классов в формате JUnit, используя CoFoJa (Contracts For Java) для задания контрактов. Испытание разработанного средства на ряде тестов показало работоспособность подхода.

Ключевые слова: автоматизированное тестирование программ; генерация тестов; частичные спецификации; контрактное программирование; анализ кода; SMT-solver

DOI: 10.14357/08696527140207

1 Введение

1.1 Автоматизация тестирования программного обеспечения

Проблема качества ПО является одной из самых острых в компьютерной индустрии. Огромные ресурсы вкладываются в различные мероприятия, направленные на повышение качества выпускаемых программных систем, так как потенциальный ущерб от сбоев ПО может быть очень велик. Одним из самых распространенных и легко реализуемых методов повышения качества

*Статья рекомендована к публикации в журнале Программным комитетом конференции “Tools & Methods of Program Analysis” («Инструменты и методы анализа программ», TMPA-2013), 10–12 октября 2013, г. Кострома, РФ.

[1]Санкт-Петербургский государственный политехнический университет, aleftina.andrianova@gmail.com

[2]Санкт-Петербургский государственный политехнический университет, vlad@icc.spbstu.ru

является тестирование. Различные технологии тестирования активно используются практически во всех компаниях, занимающихся разработкой ПО. Однако эффективность проведения тестирования определяется не только количеством разработанных тестов, но и их качеством. Зачастую качество тестов зависит только от профессиональных навыков тестировщика, в то время как требуется, чтобы качество создаваемых тестов больше зависело от разработанного ПО и спецификации требований. Поэтому в последнее время активно развивается направление программной инженерии, связанное с автоматизацией генерации тестов на основе различных проектных артефактов. В качестве таких артефактов в разных исследованиях используются спецификации требований, аннотации, контракты, исходный код ПО и т. п.

При формировании тестов перед разработчиками встают следующие основные задачи:

– обеспечение максимального покрытия тестами программы, при этом обычно рассматривается один из следующих критериев покрытия: покрытие операторов, покрытие ветвлений или покрытие путей;

– формирование тестов, покрывающих максимально возможный диапазон входных значений функций и методов;

– проверка максимально возможного числа требований, предъявляемых к программе.

В данной статье описывается разработанный авторами подход, целями которого было решение перечисленных выше задач. Подход основан на интеграции двух методов тестирования: структурного, базирующегося на анализе исходного кода программы, и функционального, использующего спецификации. Разработаны методы автоматизации создания тестов, учитывающие как внутреннее устройство программы, так и функциональные требования к ней. Для построения модели программы и извлечения трасс выполнения используются методы статического анализа. В качестве исходных требований к ПО выступают частичные спецификации в форме контрактов, позволяющие описывать требования, предъявляемые к отдельным методам и классам. Совместный анализ извлеченных трасс и контрактов позволяет формировать комплекты тестов, обеспечивающих покрытие путей программы. Подход реализован в виде инструментального средства генерации тестов на основе анализа Java-программ и системы контрактов CoFoJa [1].

1.2 Результаты

В данной работе получены следующие теоретические и практические результаты:

– разработаны принципы генерации тестов, форсирующих прохождение выбранной трассы программы;

– разработаны методы комплексирования функционального и структурного подходов, основанные на объединении анализа кода и контрактов;
– предложены методы формирования множественных тестов для обеспечения покрытия всей области определения;
– разработано инструментальное средство генерации тестов для Java-программ.

1.3 Структура статьи

Оставшаяся часть статьи организована следующим образом. Второй раздел содержит описание разработанного авторами подхода к генерации тестов. Подробно рассматривается построение модели программы, извлечение трасс исполнения и формирование системы утверждений для SMT-солверов (инструментов разрешения логических формул, Satisfiability Modulo Theories), а также генерация множественных тестов. Третий раздел посвящен описанию созданного прототипа генератора тестов. В четвертом разделе приводится информация об апробации подхода. Пятый раздел содержит обзор текущего состояния дел в области автоматизированной генерации тестов на основе анализа программы и учета контрактов. В заключении подводится итог проведенной работы, формулируются направления дальнейших исследований.

2 Предлагаемый подход

В данной работе предлагается подход к автоматизации создания модульных тестов, обеспечивающий покрытие путей программы. Далее в этой работе, не умаляя общности, будем рассматривать создание тестов для одного метода какого-либо класса. Для обеспечения покрытия путей метода необходимо для каждой возможной трассы исполнения программы сгенерировать такой набор значений аргументов метода, который форсирует прохождение программой этой трассы, т. е. необходимо так подобрать значения аргументов, чтобы при каждом возможном ветвлении выбиралась требуемая ветвь программы. Построение для заданных значений аргументов соответствующих трасс исполнения является прямой задачей. Вычисление же требуемых значений аргументов метода для удовлетворения заданной трассы — это обратная задача. В общем случае обратная задача — сложная научная NP-полная проблема, требующая решения сложных систем уравнений. В данной статье описывается способ решения этой задачи с определенными ограничениями, позволяющий вычислить требуемые значения аргументов за приемлемое время. Если ограничиться формированием модульных тестов только на основе критерия покрытия, то полученные тесты никак не будут учитывать требования, предъявляемые ко всей программе вообще или к конкретному методу в частности. Чтобы использовать информацию о требованиях, описанный способ формирования тестов на основе анализа программы может быть расширен с помощью учета спецификаций. Спецификация (или

частичная спецификация) может быть задана с помощью контрактов [2]. Анализ предусловий метода и инвариантов класса может сузить множество возможных значений параметров тестов, исключив из них не соответствующие контракту. Постусловия могут использоваться для формирования заготовки тестовых оракулов, упрощая тем самым работу разработчиков тестов.

Предлагаемый авторами подход подразумевает выполнение следующих шагов:

(1) формирование модели программы;
(2) формирование списка путей выполнения метода;
(3) преобразование каждого пути в цепочку утверждений, верных для этого пути;
(4) дополнение цепочки утверждений составляющими контрактов: предусловиями метода и инвариантами класса;
(5) разрешение системы утверждений относительно аргументов метода;
(6) формирование экземпляра/экземпляров теста на основе решения системы утверждений;
(7) формирование тестового оракула на основе постусловий метода и инвариантов класса.

Рассмотрим перечисленные шаги подробнее.

2.1 Модель анализируемой программы

Для извлечения отдельных путей выполнения программы необходимо использовать такую модель, которая, с одной стороны, содержит все необходимые данные для построения трасс, а с другой — является довольно компактной. Так как требуется извлекать информацию о динамике программы, то в качестве модели не могут использоваться структурные модели (например, абстрактное синтаксическое дерево), не содержащие информации о последовательности выполнения операторов, использование же гибридных моделей (например, абстрактного семантического графа) неоправданно, так как такие модели слишком избыточны. Среди поведенческих моделей программ для решения задачи извлечения путей наиболее подходят модели, основанные на потоке управления, такие как граф потока управления (Control Flow Graph, CFG) или модель статического однократного присваивания (Single Static Assignment, SSA). В данной работе на фазе построения ограничимся построением классического графа потока управления, элементы же однократного статического присваивания используются позже на этапе преобразования путей в цепочки утверждений.

2.2 Извлечение трасс исполнения

Извлечение трасс исполнения осуществляется с помощью анализа графа потока управления. Для этого применяется рекурсивный алгоритм поиска, выявляющий все возможные уникальные пути от начальной точки к конечной.

При этом по-разному обрабатываются узлы, соответствующие вычислительным операторам и операторам ветвления. Для вычислительных узлов в список элементов пути добавляются операторы, соответствующие этим узлам. Для узлов ветвления вида «if (condition) . . . » в список добавляется условие «condition» или «∼condition» в зависимости от выбранной при обходе графа ветви условного оператора. Таким образом, для каждого пути графа будет собран список операторов и условий, которые должны выполниться для того, чтобы данный путь был пройден.

2.3 Формирование системы утверждений

Для того чтобы по построенному пути вычислить подходящие значения аргументов, форсирующих выполнение этого пути, необходимо решить обратную задачу. Для этого требуется преобразовать полученный путь в систему уравнений и разрешить ее относительно аргументов метода. Для этого необходимо для каждого накопленного пути проделать следующие операции:

- преобразовать путь в форму однократного статического присваивания (SSA);
- все присваивания преобразовать в утверждения «равенства»;
- все условия преобразовать в соответствующие утверждения.

Для того чтобы построить такую систему уравнений, необходимо каждой конструкции сохраненной трассы поставить в соответствие алгебраическое уравнение. Для этого все присваивания преобразуются в алгебраические равенства, а сохраненные условия веток ветвления — в логические утверждения. Кроме того, необходимо перейти от переменных языка программирования, утверждения над которыми истинны только в определенном контексте выполнения, к алгебраическим переменным, истинность утверждений над которыми не меняется во времени.

Пример простой программы:

```
x = 10;
y = 20;
if (z > 5) {
 x = x + y;
}
```

Соответствующая система утверждений для пути, проходящего через ветку «then»:

```
1: x=10
2: y=20
3: z>5
4: x=x+y
```

Очевидно, что такая система утверждений неразрешима, так как, с одной стороны, `x = 10` (строка 1), а с другой стороны, `x = 30` (строка 4). Такая

коллизия обусловлена кардинальными семантическими отличиями переменных императивных языков программирования от переменных, используемых в логиках. Для разрешения этой коллизии используется известный подход, применяемый при построении модели SSA и основанный на введении дополнительных алгебраических переменных, отражающих состояния переменных языка программирования после выполнения присваивания. Такие переменные называются версиями исходной переменной, а сама операция — введением версионирования. Таким образом, после введения версионирования каждой версионированной переменной в системе уравнений значение будет присвоено только один раз. Для приведенного примера система уравнений будет следующей:

```
1: x1=10
2: y1=20
3: z1>5
4: x2=x1+y1
```

Полученная таким образом система становится алгебраически разрешимой. Она относится к классу SMT [3]. Как следствие, она может быть разрешена с помощью какого-либо SMT-солвера. Для этого необходимо:

- преобразовать систему утверждений в формат, являющийся входным для SMT-солвера;
- запустить SMT-солвер для решения системы уравнений;
- проинтерпретировать результаты работы солвера.

В случае успеха из результатов извлекаются значения аргументов метода, форсирующие выполнение указанного пути. Если солвер не смог разрешить систему уравнений, то это свидетельствует либо о наличии в программе «мертвого» кода, либо о некорректности допущений, сделанных при построении модели программы, либо о недостаточной мощности математического аппарата логики первого порядка, либо о недостаточной мощности самого солвера. Во всех случаях будет принято решение о невозможности создания теста для покрытия данного пути.

2.4 Расширение системы утверждений с помощью контрактов

Построенные тесты обеспечивают прохождение программой соответствующих трасс исполнения, при этом значения аргументов методов расположены в области допустимых значений произвольно, в соответствии с правилами вывода решений выбранного SMT-солвера [3]. Это означает, что сгененрированные значения аргументов могут нарушать контракты методов, определяющиеся с помощью предусловий, постусловий и инвариантов [2]. Конкретно в данном случае могут нарушаться предусловия методов или инварианты классов. Решение этой проблемы заключается в расширении построенной системы утверждений с помощью инвариантов классов и предусловий методов. Так как контракты уже

задаются в форме утверждений логики первого порядка, то преобразование их в утверждения для SMT-солвера не представляет никакого труда.

В результате после решения расширенной системы утверждений будут находиться аргументы метода, удовлетворяющие контракту анализируемого метода.

2.5 Генерация тестовых оракулов

Для генерации полноценной системы тестов кроме значений аргументов методов необходимо также формировать тестовые оракулы, проверяющие корректность выполнения тестов. Постусловия, используемые в контрактном программировании, являются частью спецификаций методов и задают свойства, гарантируемые методом после его завершения в случае выполнения предусловий. В данной работе предлагается генерировать основу тестовых оракулов на базе постусловий методов и инвариантов классов, как это сделано в [4].

Заданные в форме логических утверждений постусловия и инварианты транслируются в эквивалентные конструкции на целевом языке программирования и интегрируются в созданные тесты. При этом у тестировщика остается возможность расширить сгенерированные автоматически оракулы дополнительными проверками.

2.6 Формирование множественных тестов

Представленная технология формирования тестов обеспечивает генерацию по одному набору тестов для каждого класса эквивалентности. Этого достаточно для того, чтобы по одному разу протестировать все пути в графе потока управления. Однако на практике часто имеет смысл иметь несколько тестов для каждого класса эквивалентности, так как необходимо проверить не только сам факт прохождения программы по заданной трассе, но и другие свойства.

Например, часто требуется проверять корректность работы программы на граничных значениях интервалов, правильность отработки начала и конца цикла и т. п. Для этого необходимо для каждого класса эквивалентности генерировать множество тестов, обеспечивающих определенное покрытие области допустимых значений аргументов в соответствии с выбранной эвристикой.

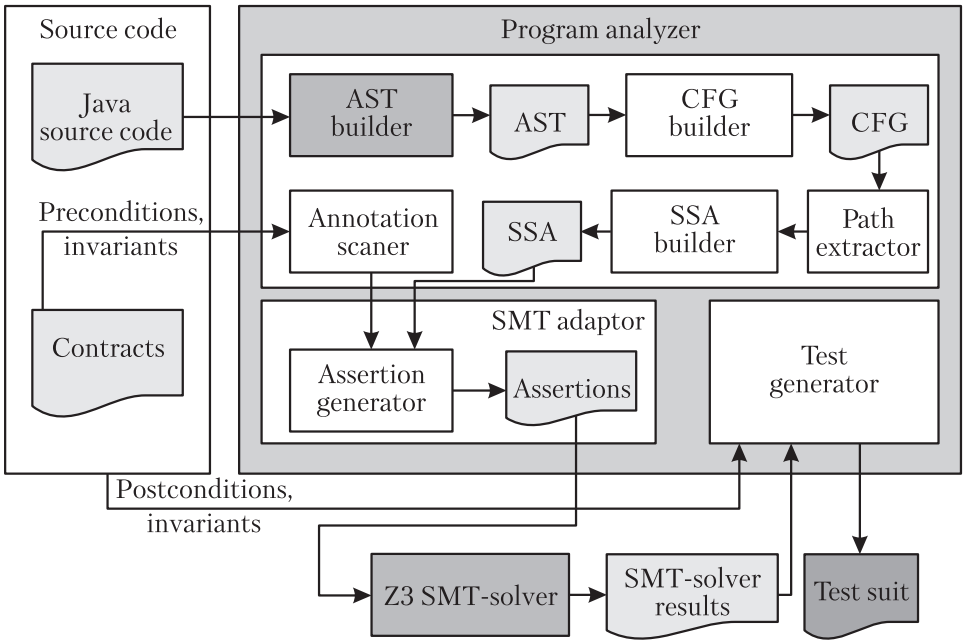
В данной работе реализован метод формирования множественных тестов, основанный на последовательной генерации новых тестов для классов эквивалентности за счет вычисления области допустимых значений для аргументов метода. Исходя из количества тестов, которые должны быть сгенерированы для каждого класса эквивалентности (задано пользователем), строится равномерное покрытие n-мерного пространства аргументов функции на основе ограничений области допустимых значений, заданных в контрактах. Для каждого набора параметров генерируется свой модульный тест в результирующем тестовом наборе. Алгоритм формирования равномерного покрытия пространства аргументов с ограничениями имеет итеративный характер, а его подробное описание выходит за рамки настоящей статьи.

В дальнейшем авторы предполагают использовать более сложные эвристики для формирования тестовых наборов, более чувствительных к стандартным ошибкам, встречающимся в программах.

3 Описание созданного прототипа

Разработанный подход был реализован в виде инструментального средства генерации тестов для языка Java. В качестве системы задания контрактов используется CoFoJa [1]. В качестве SMT-солвера применяется солвер Z3 [3], разработанный Microsoft Research, тесты генерируются в формате JUnit. Общая архитектура системы вместе с артефактами изображена на рисунке. Блоками белого цвета изображены модули, разработанные авторами, серым цветом отмечены сторонние компоненты.

На вход системы подается файл, содержащий исходные коды исследуемой программы на языке Java. Построитель AST формирует абстрактное синтаксическое дерево, которое является входом для построителя графа потока управления (CFG builder). Абстрактное синтаксическое дерево анализируется, и на основе семантики языка Java строится граф потока управления. Извлечение трасс



Общая схема системы генерации тестов

исполнения (Path extractor) осуществляется с помощью анализа графа потока управления. Полученные пути преобразуются к форме однократного статического присваивания построителем SSA (SSA builder). Преобразование пути в форму SSA с версионированием позволяет рассматривать его как систему алгебраических уравнений. Полученная система утверждений в совокупности с инвариантами класса и предусловиями метода приводится к формату SMT-LIB для внешнего SMT-солвера Z3 и передается ему для решения. Генерация тестов осуществляется модулем Test generator, который использует результаты работы солвера, при этом генерация тестовых оракулов, проверяющих корректность выполнения тестов, осуществляется на основе анализа постусловий метода и инвариантов класса.

Результатом работы системы являются сгенерированные модульные тесты, обеспечивающие определенный заданный уровень покрытия исходного кода.

4 Экспериментальные исследования

Было проведено испытание разработанного инструментального средства на тестовом наборе программ. Набор состоял из искусственных тестов и нескольких реальных программ, содержащих различные конструкции языка Java: составные выражения, простые и сложные ветвления, циклы и т. п. Искусственные тесты были призваны проверить корректность функционирования разных возможностей, заложенных в разработанном анализаторе. Реальные Java-программы использовались для проверки анализатора в реальных условиях функционирования.

На всех искусственных и реальных примерах были получены результаты, соответствующие ожидаемым, а сгенерированные тесты действительно обеспечивали проверку всех путей исследуемых программ. Эксперименты с генерацией множественных тестов также показали работоспособность подхода.

5 Обзор существующих подходов в области генерации тестов

Различные аспекты автоматизации тестирования ПО и подходы к организации статического и динамического анализа являются популярными направлениями исследований. Существует целый ряд коммерческих и свободно распространяемых средств автоматизации тестирования, фреймворков для организации разработки систем для генерации тестов. Многие из них используют в качестве основы подход синтетического структурного тестирования. В этом случае после первого случайно выбранного теста остальные тесты генерируются автоматически так, чтобы обеспечить покрытие еще не покрытых ранее элементов кода. Для выбора подходящих тестовых данных используются решатели, учитывающие символическую информацию об ограничениях на данные, отделяющие прошедшие тесты от еще не покрытого кода, а для построения нужных после-

довательностей воздействий — случайная генерация, направляемая как этой же символической информацией, так и некоторыми эвристическими абстракциями, уменьшающими пространство состояний проверяемой системы. В рамках этого подхода интегрируются статический анализ кода, символическое исполнение, структурное тестирование и дедуктивный анализ, выполняемый решателями. Одними из представителей этого класса являются Microsoft Pex and Moles, два исследовательских проекта, направленных на повышение производительности тестировщиков и качества кода.

Microsoft Pex [5, 6] — набор инструментов, выполняющих анализ тестируемого кода и подбирающий параметры, которые позволяют покрыть максимально возможный объем кода. Microsoft Pex работает в петле обратной связи: код выполняется несколько раз и производится анализ поведения программы на уровне промежуточного языка .NET CIL (Common Intermediate Language) с помощью мониторинга потока управления и потока данных. После каждого запуска Pex выбирает ветвь, которая не была покрыта ранее, создает систему ограничений, которая описывает, как достичь этой ветви, использует решатель Z3 для определения новых значений входных данных, которые удовлетворяют ограничениям, если таковые существуют. Код выполняется снова с новым набором входных значений, и процесс повторяется. В результате работы создается отчет, который можно проанализировать вручную, и набор unit-тестов для дальнейшего использования. Microsoft Moles [6, 7] — фреймворк, также разработанный в лаборатории Microsoft Research и позволяющий разработчикам создавать тестовые заглушки и избегать непосредственного вызова методов в .NET. Moles расширяет Pex за счет формирования тестового окружения и гибкого манипулирования драйверами и заглушками. Существует ряд ограничений при использовании указанных инструментов. Во-первых, невозможно обнаруживать ошибки, которые вносятся при параллельном выполнении задач. Во-вторых, указанные средства не могут справиться с недетерминизмом. Это приводит к разным результатам при каждом запуске Pex. В-третьих, Pex и Moles не могут быть использованы для анализа кода, который не работает под управлением .NET. Множество поддерживаемых языков ограничено только языком C#. Возможность использования внешнего солвера, отличного от Z3, также не предусмотрена.

Еще один подход к автоматизации модульного тестирования был предложен Engel и Hähnle. В работе [8] рассматривается метод автоматической генерации тестов для JAVA CARD, базирующийся на формальной проверке выполнения тестируемого кода. В его основе лежат два подхода к тестированию: по методу «белого» и «черного» ящика. Автономные модульные тесты в формате JUnit генерируются автоматически. Verification-based test generation (VBT) использует полную информацию, содержащуюся в формальной спецификации и лежащую в основе реализации тестируемого кода (implementation under test, IUT). В качестве языка описания спецификации используется Java Modeling Language (JML). При этом полная функциональная спецификация тестируемого кода не требуется, поскольку генерация тестов реализуется на основе символического

исполнения, а заданные в пред- и постусловиях ограничения требуются только для формирования тестовых оракулов.

Система Kiasan [9, 10] представляет собой механизм символического исполнения для последовательных Java-программ, основанный на фреймворке Bogor, использующем технологию проверки модели (model checking). Предложенный подход был позднее расширен в фреймворке KUnit [10], который автоматически генерирует тесты для контрактно-аннотированых методов и отображает множество объектов (heap objects), получаемых и возвращаемых методами, что может быть использовано разработчиками для анализа сложных методов и диагностики причин ошибок в программе. Фреймворк KUnit использует тестирование по методу черного ящика с помощью генерации входных значений на основе символического исполнения, постусловия используются в качестве тестовых оракулов. При наличии формальной спецификации подход, используемый в KUnit, может реализовывать такие техники тестирования, как разделение на классы эквивалентности и анализ граничных значений. В общем случае традиционное тестирование обычно не основывается на формальной спецификации, в то время как KUnit требует формального описания требований.

Инструмент Unit Meister [11], разработанный компанией Microsoft, использует символическое исполнение для анализа трасс программы. Подход во многом схож с Kiasan, однако существует ряд отличий. Unit Meister включает функциональные символы для композиционной проверки, в то время как Kiasan допускает только исходные символы, а композиционная проверка осуществляется с помощью спецификаций. И если Kiasan является полностью автоматическим, то Unit Meister требует от пользователя указания области определения параметров.

Symstra [12] — еще одно средство символического исполнения для создания минимальной последовательности вызовов публичных методов для проверки класса. Symstra использует примитивные символические значения, конкретные структуры множеств и предполагаемые состояния для генерации неизоморфных конечных состояний. Для генерации подмножества возможных предварительных состояний в Symstra применяются последовательности вызовов публичных методов. В разработанном прототипе предварительные состояния задаются предусловиями и инвариантами. Пользователь может изменять предусловия и инварианты, более точно определяя их значения.

Одним из инструментов, ориентированных на использование контрактных спецификаций в процессе создания тестов, является AutoTest framework [4]. Он автоматизирует процесс тестирования ПО, опираясь на программы, содержащие инструменты их собственной проверки, в форме контрактов для классов и отдельных методов. Предусловия и инварианты позволяют ограничить множество входных данных для тестирования, постусловия преобразуются в тестовые оракулы. Основная особенность AutoTest заключается в способе генерации тестов. Тестовые последовательности формируются случайно, опираясь только на предсуловия методов и инварианты классов. Успешность прохождения тестов анализируется в тестовых оракулах. Отдельно стоит отметить механизм

Test Extraction, который автоматически создает тесты по результатам отказов программы. Основными ограничениями подхода являются отсутствие гарантий покрытия путей (из-за сугубо случайной генерации тестов), а также поддержка только языка Eiffel и среды EiffelStudio.

Все рассмотренные подходы подтверждают эффективность интеграции различных методологий процесса тестирования на практике. Тем не менее, несмотря на достигнутые успехи, каждый из имеющихся подходов использует лишь часть имеющегося потенциала, ограничен анализом программ, написанных на каком-то определенном языке программирования, и не предоставляет единой интеграционной платформы для всего многообразия различных техник верификации ПО. Кроме того, большинство из рассмотренных подходов являются академическими и не применимы для анализа сложных программных систем [13, 14].

6 Заключение

В результате проведенного исследования разработан подход к генерации тестов для языка Java, обеспечивающих покрытие трасс исполнения методов. Разработанные методы с помощью применения SMT-солвера генерируют параметры модульных тестов, форсирующие реализацию заданных трасс исполнения. Учет контрактов методов позволяет, с одной стороны, с помощью постусловий частично автоматизировать генерацию тестовых оракулов, а с другой — с помощью предусловий ограничивать генерацию параметров тестов. Применение разработанного на основе подхода прототипа на наборе тестовых примеров показало полную работоспособность предложенных методов. Основными направлениями развития теоретических и практических исследований являются:

- разработка новых алгоритмов генерации множественных тестов, более эффективно распределяющих значения переменных по области определения;
- совершенствование анализатора (прототипа) с целью обеспечения анализа более широкого класса Java-программ;
- расширение функциональных возможностей и реализация подхода генерации модульных тестов для программ, написанных на других языках программирования.

Литература

1. Contracts for Java. https://code.google.com/p/cofoja.
2. *Meyer B.* Design by contract. Advances in object-oriented software engineering / Eds. D. Mandrioli and B. Meyer. — Upper Saddle River, NJ, USA: Prentice Hall, 1991. P. 1–50.
3. Z3: An efficient SMT Solver. http://z3.codeplex.com.
4. *Meyer B., Ciupa I., Leitner A., Fiva A., Yi Wei, Stapf E.* Programs that test themselves // IEEE Comp., 2009. Vol. 42. No. 9. P. 46–55.

5. *Tillmann N., Halleux J.* Pex: White box test generation for .NET // TAP'08: 2nd Conference (International) on Tests and Proofs Proceedings. — Berlin, Heidelberg: Springer-Verlag, 2008. P. 134–153.
6. Pex and Moles — isolation and white box unit testing for .NET. http://research.microsoft.com/en-us/projects/pex.
7. Unit testing with Microsoft Moles: Tutorial for lightweight test stubs and detours for .NET applications. Ver. 0.93, 2010. http://research.microsoft.com/en-us/projects/pex/molestutorial.pdf.
8. *Engel C., Hähnle R.* Generating unit tests from formal proofs // Tests and proofs / Eds. Yu. Gurevich, B. Meyer. Lecture notes in computer science ser. — Berlin–Heidelberg: Springer-Verlag, 2007. Vol. 4454. P. 169–188.
9. *Deng X., Lee J., Robby.* Bogor/Kiasan: A k-bounded symbolic execution for checking strong heap properties of open systems // ASE'06: 21st IEEE/ACM Conference (International) on Automated Software Engineering Proceedings. — Washington, DC, USA: IEEE Computer Society, 2006. P. 157–166.
10. *Deng X., Robby, Hatcliff J.* Kiasan/KUnit: Automatic test case generation and analysis feedback for open object-oriented systems // Testing: Academic and Industrial Conference Practice and Research Techniques (TAICPART) Proceedings. — Washington, USA: IEEE Computer Society, 2007. P. 3–12.
11. *Tillmann N., Schulte W.* Parameterized unit tests with unit meister // 10th European Software Engineering Conference and 13th ACM Sigsoft Symposium (International) Foundations of Software Engineers (ESEC/Sigsoft FSE) Proceedings. — New York, NY, USA: ACM Press, 2005. P. 241–244.
12. *Xie T., Marinov D., Schulte W., Notkin D.* Symstra: A framework for generating object-oriented unit tests using symbolic execution // Tools and algorithms for the construction and analysis of systems / Eds. N. Halbwachs, L. D. Zuck. Lecture notes in computer science ser. — Berlin–Heidelberg: Springer-Verlag, 2005. Vol. 3440. P. 365–381.
13. *Robschink T., Snelting G.* Efficient path conditions in dependence graphs // ICSE'02: 24th Conference (International) on Software Engineering Proceedings. — New York, NY, USA: ACM Press, 2002. P. 478–488.
14. *Henkel J., Diwan A.* Discovering algebraic specifications from Java classes // ECOOP 2003 — object-oriented programming / Ed. L. Cardelli. Lecture notes in computer science ser. — Berlin–Heidelberg: Springer-Verlag, 2003. Vol. 2743. P. 431–456.

Поступила в редакцию 29.01.14

SOURCE CODE AND PARTIAL SPECIFICATIONS ANALYSIS FOR AUTOMATED GENERATION OF UNIT TESTS*

A. Andrianova and V. Itsykson

St. Petersburg State Polytechnical University, 29 Polytechnicheskaya Str., St. Petersburg 195251, Russian Federation

Abstract: Low level of software quality is one of the main problems of software engineering. Automated testing is one of the most effective strategies used to improve the quality of the software. This paper describes a technique of automated unit tests creation combining both functional and structural approaches of software testing. In this method, information extracted from the original program is used to ensure test coverage of program paths. Partial specifications given in the form of contracts are used to form test oracles and to distribute parameters of tests on definition domain. The developed approach was implemented as a tool that analyzes Java programs and generates test cases for class methods in JUnit format, using CoFoJa to specify the contracts. Designed tool testing on a number of test cases showed efficiency of the approach.

Keywords: automated software testing; unit test generating; contract-based programming; code analysis; SMT-solver

DOI: 10.14357/08696527140207

References

1. Contracts for Java. Available at: https://code.google.com/p/cofoja/ (accessed March 18, 2014).
2. Meyer, B. 1991. Design by contract. *Advances in object-oriented software engineering*. Eds. D. Mandrioli and B. Meyer. Upper Saddle River, NJ, USA: Prentice Hall. 1–50.
3. Z3: An efficient SMT solver. Available at: http://z3.codeplex.com (accessed March 18, 2014).
4. Meyer, B., I. Ciupa, A. Leitner, A. Fiva, W. Yi, and E. Stapf. 2009. Programs that test themselves. *IEEE Comp.* 42(9):46–55.
5. Tillmann, N., and J. Halleux. 2008. Pex: White box test generation for .Net. *TAP'08: 2nd Conference (International) on Tests and Proofs Proceedings*. Berlin, Heidelberg: Springer-Verlag. 134–153.
6. Pex and Moles — isolation and white box unit testing for .NET. Available at: http://research.microsoft.com/en-us/projects/pex/ (accessed March 18, 2014).

*The article is published in accordance to the recommendation of the Program Committee of the International Conference International Conference "Tools & Methods of Program Analysis 2013, TMPA-2013."

7. Unit testing with Microsoft Moles. Tutorial for lightweight test stubs and detours for .NET applications. Available at: http://research.microsoft.com/en-us/projects/pex/molestutorial.pdf (accessed March 18, 2014).
8. Engel, C., and R. Hähnle. 2007. Generating unit tests from formal proofs. *Tests and proofs*. Eds. Yu. Gurevich and B. Meyer. Berlin–Heidelberg: Springer-Verlag. Lecture notes in computer science ser. 4454:169–188.
9. Deng, X., J. Lee, and Robby. 2006. Bogor/Kiasan: A *k*-bounded symbolic execution for checking strong heap properties of open systems. *21st IEEE/ACM Conference (International) on Automated Software Engineering (ASE'06) Proceedings.* IEEE Computer Society, Washington, DC, USA. 157–166.
10. Deng, X., Robby, and J. Hatclif. 2007. Kiasan KUnit: Automatic test case generation and analysis feedback for open object-oriented systems. *Testing: Academic and Industrial Conference Practice and Research Techniques (TAICPART) Proceedings*. Washington, DC, USA: IEEE Computer Society. 3–12.
11. Tillmann, N., and W. Schulte. 2005. Parameterized unit tests with unit meister. *ESEC/SIGSOFT FSE*. ACM. 241–244.
12. Xie, T., D. Marinov, W. Schulte, and D. Notkin. 2005. Symstra: A framework for generating object-oriented unit tests using symbolic execution. *Tools and algorithms for the construction and analysis of systems*. Eds. N. Halbwachs and L. D. Zuck. Lecture notes in computer science ser. Berlin–Heidelberg: Springer-Verlag. 3440:365–381.
13. Robschink, T., and G. Snelting. 2002. Efficient path conditions in dependence graphs. *ICSE'02: 24th Conference (International) on Software Engineering Proceedings*. New York, NY, USA: ACM Press. 478–488.
14. Henkel, J., and A. Diwan. 2003. Discovering algebraic specifications from Java classes. *ECOOP 2003: Object-oriented programming*. Ed. L. Cardelli. Lecture notes in computer science ser. Darmstadt, Germany. 2743:431–456.

Received January 29, 2014

Contributors

Andrianova Aleftina A. (b. 1991) — MS in technology, St. Petersburg State Polytechnical University, 29 Polytechnicheskaya Str., St. Petersburg 195251, Russian Federation; aleftina.andrianova@gmail.com

Itsykson Vladimir M. (b. 1973) — Candidate of Science (PhD) in technology, Associate Professor, St. Petersburg State Polytechnical University, 29 Polytechnicheskaya Str., St. Petersburg 195251, Russian Federation; vlad@icc.spbstu.ru

ОБНАРУЖЕНИЕ ГОНОК В JAVA-ПРОГРАММАХ С ПРИМЕНЕНИЕМ СИНХРОНИЗАЦИОННЫХ КОНТРАКТОВ*

Д. И. Цителов[1], *В. Ю. Трифанов*[2]

Аннотация: Состояние гонки (data race) возникает в многопоточной программе, когда несколько потоков одновременно обращаются к одному и тому же разделяемому участку памяти, где хотя бы одно обращение — запись. Состояния гонки трудновоспроизводимы и могут приводить к повреждению глобальных структур данных, поэтому исследования в области автоматического поиска гонок ведутся уже более 20 лет. В данной статье рассматривается вопрос повышения производительности динамического поиска гонок в Java-программах без существенной потери точности. Для решения этой задачи используются синхронизационные контракты — частичные спецификации поведения программных методов и классов в многопоточной среде. Применение контрактов позволяет исключать из анализа не интересные с точки зрения поиска гонок части целевого приложения (например, сторонние библиотеки). В статье рассматриваются преимущества и ограничения подхода, описывается язык спецификации контрактов и некоторые технические детали реализации.

Ключевые слова: многопоточность; состояние гонки; динамический анализ; автоматическое обнаружение ошибок

DOI: 10.14357/08696527140208

1 Введение

С развитием многоядерных и многопроцессорных систем все большее количество программ создаются многопоточными. Использование нескольких потоков выполнения может приводить к дополнительным ошибкам, связанным с некорректной организацией взаимодействия этих потоков. Такие ошибки сложно искать вручную, поскольку чередование операций в потоках обладает высокой степенью неопределенности и программисту нужно полностью просчитать все возможные ситуации. Одной из типичных ошибок многопоточных программ является наличие в них состояний гонки (data races). Они возникают, когда несколько потоков без должной синхронизации обращаются к одному и тому же разделяемому участку памяти, где хотя бы одно обращение — запись [1]. Дополнительная трудность с обнаружением гонок состоит в том, что они, как правило,

*Статья рекомендована к публикации в журнале Программным комитетом конференции “Tools & Methods of Program Analysis” («Инструменты и методы анализа программ», TMPA-2013, 10–12 октября 2013, г. Кострома, РФ).

[1]ООО «Эксперт-Система», tsitelov@acm.org

[2]ООО «Эксперт-Система СЗ», vitaly.trifanov@gmail.com

не приводят к немедленному сбою и отказу программы. Напротив, приложение продолжает работать с поврежденными глобальными структурами данных, что приводит к труднообъяснимым эффектам впоследствии.

За несколько последних десятков лет было разработано несколько статических и динамических подходов к обнаружению гонок. *Статические* детекторы [2–4] анализируют все пути исполнения программы, не требуют ее запуска, но обладают низкой точностью. *Динамические* [5–12] анализируют программу во время ее работы, но ограничены лишь фактическим путем выполнения программы. Хотя динамические алгоритмы и обладают полной информацией о конкретном пути выполнения программы, на практике их точность ограничена накладными расходами, которые они вносят. Проблема достижения высокой точности обнаружения гонок и отсутствия ложных срабатываний (*false positives*) при сохранении приемлемого уровня потребления аппаратных ресурсов является краеугольной технической проблемой динамического анализа.

Данная работа предлагает способ понижения накладных расходов на динамическое обнаружение гонок в Java-программах без потери точности. Основная идея заключается в сокращении области программы, где отслеживаются все операции синхронизации. Чтобы это не привело к потере информации о синхронизации между потоками и, как следствие, к ложным срабатываниям, вводится понятие синхронизационного контракта, которое позволяет в достаточной степени описать поведение исключенного кода в многопоточной среде. Далее во время работы детектор распознает эти контракты и, встретив методы, соответствующие им, обрабатывает эти методы как высокоуровневые синхронизационные события. Такой подход позволяет восполнить неотслеженные операции синхронизации и обеспечить высокую точность поиска.

Дальнейшая часть статьи организована следующим образом. Во второй главе описывается отношение happens-before и приводится формальное определение понятия гонки в терминах спецификации Java, после чего описывается точный алгоритм поиска гонок, основанный на отслеживании этого отношения. Третья глава акцентируется на проблеме обеспечения точности динамического поиска гонок и одновременно приемлемого уровня накладных расходов. В четвертой главе излагается концепция синхронизационных контрактов, в пятой главе — ее реализация в разработанном авторами динамическом детекторе гонок jDRD. Шестая глава посвящена преимуществам и недостаткам подхода и содержит некоторые экспериментальные результаты. В заключении подводится итог и указываются дальнейшие возможные направления работы. В приложении описан язык описания контрактов, используемый в jDRD.

2 Happens-before-алгоритм поиска гонок

Отношение happens-before было предложено Лесли Лампортом в работе [13] для установления частичного порядка на множестве всех событий распределенной системы, в которой единственный способ взаимодействия между участниками —

это обмен сообщениями. Данный подход хорошо переносится на многопоточные системы, если рассматривать каждый поток как отдельного участника системы, а операции синхронизации — как передачу сообщений. Язык программирования Java был одним из первых языков с собственной архитектурно-независимой моделью памяти, определяющей семантику взаимодействия потоков через отношение happens-before и распространяющей это отношение на все операции в программе.

Для этого в спецификации Java [14] сначала вводится отношение *синхронизированности* (synchronized-with) — полное отношение порядка на множестве всех операций синхронизации в программе. Так:

- освобождение монитора *синхронизировано* с его последующими захватами;
- запись volatile-переменной синхронизирована с ее последующими чтениями;
- запуск потока *синхронизирован* с первым действием в потоке;
- последнее действие в потоке T_1 *синхронизировано* с любым действием в другом потоке, случившемся после того, как тот обнаружил, что T_1 завершился;
- прерывание потока T_1 *синхронизировано* с любым действием в другом потоке, которое обнаружит, что T_1 получил сигнал прерывания.

События слева (например, освобождение монитора) будем называть *отправкой*, а справа (захват монитора) — *приемом* синхронизационного сообщения. Обратим внимание, что во всех случаях с синхронизационным сообщением естественным образом ассоциирован уникальный объект — монитор, volatile-переменная или объект потока. Поэтому далее будем считать, что с каждой парой синхронизированных событий связан некий уникальный *синхронизационный объект*.

Объединение отношения синхронизированности с естественным темпоральным отношением порядка в рамках операций одного потока дает частичное отношение порядка — отношение *предшествования* (happens-before), определенное уже на множестве всех операций в программе. Факт «событие x произошло перед событием y» записывается как $\mathrm{hb}\,(x,\, y)$. Само отношение определяется следующим образом:

- если x и y — операции в одном потоке и x произошло раньше y, то $\mathrm{hb}\,(x,\, y)$;
- завершение конструктора объекта предшествует запуску его финализатора;
- если события x и y синхронизированы, то $\mathrm{hb}\,(x,\, y)$;
- транзитивность: если $\mathrm{hb}\,(x,\, y)$ и $\mathrm{hb}\,(y,\, z)$, то $\mathrm{hb}\,(x,\, z)$.

Отслеживание отношения happens-before позволяет определить, получил ли некоторый поток информацию о действиях другого потока — например, изменения в разделяемой памяти, новые значения переменных и т. д.

Наконец, спецификация Java формально определяет понятие гонки: два обращения x и y к разделяемой переменной из различных потоков, хотя бы одно

из которых — запись, образуют гонку, если ни одно из них не предшествует другому:

$$\mathrm{DR}\,(x, y) \Leftrightarrow !\mathrm{hb}\,(x, y) := !\mathrm{hb}\,(y, x)\ .$$

Таким образом, для точного обнаружения гонок в программе достаточно отслеживать конфликтующие обращения различных потоков к разделяемым переменным, не упорядоченные отношением happens-before. Для этого традиционно используются *векторные часы* (vector clock)[1], предложенные в работе [15]. Векторные часы представляют собой динамический массив неотрицательных чисел, равный по размеру количеству потоков в программе. Над векторными часами определена операция слияния:

$$\mathrm{merge}\,(V_1, V_2):\ \text{for each } i \in [1, \ldots, n]\ V_1[i] := \max(V_1[i], V_2[i])\,.$$

Каждый поток хранит свои векторные часы, отражающие его знания о системе: i-я компонента его часов равна последней компоненте i-го потока, которые он наблюдал[2]. Компонента, соответствующая ему самому, называется *собственной* компонентой. Изначально собственная компонента часов потока проинициализирована единицей, а остальные — нулями. Перед каждой операцией синхронизации x в потоке T_1 собственная компонента часов потока T_1 увеличивается на единицу, а часы загружаются в часы соответствующего синхронизационного объекта. Когда в другом потоке T_2 происходит событие y такое, что x синхронизированно с y, в часы потокаT_2 загружаются часы синхронизационного объекта. Это позволяет отследить отношение synchronized-with.

Для отслеживания отношения happens-before нужно проассоциировать векторные часы с каждой разделяемой переменной в программе. Когда поток T_1 обращается к такой переменной, он покомпонентно сравнивает свои часы с часами переменной. Если есть компонента часов потока, которая не превосходит соответствующей компоненты часов переменной, обнаружена гонка. Далее поток загружает свои часы в часы переменной.

Если в программе N потоков, то хранение одних векторных часов требует $O(N)$ памяти и основные операции над ними также имеют сложность $O(N)$. В работе [10] показано, что для отслеживания отношения happens-before вместо полных часов переменной достаточно хранить номер и собственную компоненту потока, который последним обращался к этой переменной. Таким образом, значительная часть операций будет требовать $O(1)$ и снизится количество потребляемой памяти, что позволило алгоритму happens-before достичь уровня производительности менее точных динамических алгоритмов.

[1] Эквивалентное название — *логические часы* (logical clocks).

[2] Не умаляя общности, предполагается, что все потоки пронумерованы от 1 до n.

3 Точность поиска и накладные расходы

К сожалению, точечные оптимизации алгоритма, хотя и производят безусловный положительный эффект (что подтверждает множество исследований), не могут обеспечить приемлемый уровень накладных расходов на приложениях, состоящих хотя бы из нескольких тысяч классов и использующих 10–20 потоков[3]. Подавляющее большинство исследовательских работ останавливается на прототипе детектора и модельном тестировании. Подробный обзор состояния предметной области можно найти в работах [16, 17]. Авторам удалось обнаружить лишь два готовых к использованию детектора (IBM MSDK [11] и TSan [12]), но попытки использовать их на крупных приложениях приводили либо к немедленному отказу, либо к переполнению памяти. О схожих проблемах сообщают также авторы работы [8].

Для обеспечения точности поиска нужно отслеживать все операции обращения к разделяемым данным и все операции синхронизации. Первые можно ограничить путем отсечения областей кода, в которых гонки не представляют интереса, — например, сторонние библиотеки или стандартные Java-классы. Для большинства разрабатываемых программных систем решение об использовании сторонних компонент принимается на стадии проектирования, когда уже известны требования по надежности и безотказности системы, поэтому есть смысл считать, что сторонние компоненты обладают достаточной степенью надежности, и концентрироваться на обнаружении ошибок в непосредственно разрабатываемом программном коде и проверке корректности использования сторонних компонент.

Однако исключить таким же образом операции синхронизации нельзя, поскольку их пропуск приведет к неполучению информации о передаче отношения happens-before и, как следствие, к ложным срабатываниям — детектор будет сигнализировать о гонках, которых в действительности не произошло. Количество этих операций велико и экспоненциально растет с увеличением числа потоков в программе, что в совокупности с необходимостью отслеживать операции обращения к разделяемым данным в итоге и приводит к описанным эффектам.

4 Синхронизационные контракты

Принцип инкапсуляции в объектно-ориентированном подходе к программированию, которому следует Java, предполагает, что использование объекта осуществляется посредством вызова его публичных методов. Поэтому в идеале достаточно иметь возможность описать правила использования методов исключенных классов в многопоточной среде. Возможны следующие варианты:

(1) метод не потокобезопасен, т. е. его одновременное использование несколькими потоками не предусмотрено и требует внешней синхронизации;

[3]Обратим внимание, что характеристики многих промышленных программных систем могут в десятки и тысячи раз превосходить указанные.

(2) метод потокобезопасен; он может быть вызван (и в некотором смысле предназначен для этого) несколькими потоками одновременно без внешней синхронизации.

Вызовы методов первого типа необходимо отслеживать и обрабатывать по алгоритму happens-before как обращения к объекту-владельцу метода на чтение, если метод немодифицирующий, или на запись в противном случае. Разработанный авторами детектор jDRD трактует по умолчанию все обращения как модифицирующие и предоставляет возможность на уровне конфигурации указать, что некоторые конкретные методы являются немодифицирующими.

С вызовами методов второго типа немного сложней. Вообще говоря, их можно игнорировать, поскольку они предназначены для многопоточного использования. Однако именно в эту группу методов попадают те, которые обеспечивают синхронизацию между потоками. Как правило, такие методы содержатся в классах, созданных как средство обеспечения корректного взаимодействия потоков, что четко декларировано в их описании. Так, начиная с версии 1.5 в Java появились высокоуровневые средства синхронизации, отличные от захватов/блокировок мониторов и volatile-переменных. В основном они содержатся в пакете java.util.concurrent и его подпакетах. Например, там есть потокобезопасная хеш-таблица ConcurrentHashMap, которая гарантирует, что вызов метода put по некоторому ключу *предшествует* последующему вызову метода get на том же объекте по тому же ключу [18].

Более формально это означает, что между событиями вызова метода put в первом потоке и метода get во втором потоке есть цепочка событий, через которые передается отношение happens-before. Если детектор отслеживает все операции синхронизации, то он обнаружит эту цепочку и вычислит, что эти два вызова методов также упорядочены отношением happens-before.

Теперь предположим, что необходимо исключить класс ConcurrentHashMap из области анализа, т. е. не отслеживать в нем операции синхронизации. В этом случае детектор не получит информации о том, что вызов put предшествует вызову get. Следовательно, ему ее нужно явно сообщить. Отметим, что для этого не подходят аннотации, поскольку в случае библиотечных классов нет возможности модифицировать исходный код. Авторами был разработан метод конфигурирования на базе xml (extensible markup language).

Рассмотрим два метода: метод $f(P_{11}, \ldots, P_{1n})$ объекта O_1 и метод $g(P_{21}, \ldots, P_{2m})$ объекта O_2, где $n, m \geq 0$. Объект O_1 будем называть объектом-владельцем метода f, а O_2 — объектом-владельцем метода g. Между этими методами может быть *примитивная явная связь* одного из трех типов:

(1) связь «владелец–владелец»: $O_1 = O_2$, т. е. методы принадлежат одному объекту;

(2) связь «владелец–параметр»: $n > 0$, $\exists\, i \in [1, \ldots, n]$: $O_2 = P_{1i}$ или $m > 0$, $\exists\, j \in [1, \ldots, m]$: $O_1 = P_{2j}$, т. е. параметр одного метода является объектом-владельцем другого метода;

```
<Sync>
 <Links>
   <Link send="owner" receive="owner"/>
   <Link send="param" send-number="0" receive="param" receive-number="0"/>
 </Links>
 <Send>
   <MethodCall owner="java.util.concurrent.ConcurrentMap" name="put"
descriptor="(Ljava/lang/Object;Ljava/lang/Object;) Ljava/lang/Object;"/>
 </Send>
 <Receive>
   <MethodCall owner="java.util.concurrent.ConcurrentMap" name="get"
descriptor="(Ljava/lang/Object;)Ljava/lang/Object;"/>
 </Receive>
</Sync>
```

Рис. 1 Пример синхронизационного контракта

(3) связь «параметр–параметр»: $n, m > 0$, $\exists\ i \in [1, \ldots, n]$, $j \in [1, \ldots, m]$: $P_{1i} = P_{2j}$, т. е. i-й параметр метода f и j-й параметр метода g являются одним и тем же объектом.

Будем называть *явной связью* комбинацию любого количества примитивных связей.

Будем называть *синхронизационным контрактом* описание пары *явно связанных* методов, которые, будучи вызванными в определенном порядке, гарантируют синхронизацию потоков. Рассмотрим пример с классом ConcurrentHashMap. Синхронизационный контракт на методы put и get — это явная связь, образованная из связей первого (речь идет о методах одной и той же map) и третьего (должен быть один и тот же ключ — первый параметр каждого метода) типа. Описание этого контракта на языке конфигурирования jDRD представлено на рис. 1. Подробное описание языка см. в приложении.

В качестве примера примитивной связи второго типа можно привести контракт метода execute класса Executor, обеспечивающий асинхронное выполнение задачи. Он принимает в качестве параметра объект типа Runnable и впоследствии вызывает его метод run в другом потоке. Спецификация метода execute гарантирует, что его вызов *предшествует* последующему вызову метода run объекта, переданного в метод execute в качестве параметра.

В следующем разделе будет представлен созданный авторами детектор jDRD и показана реализация синхронизационных контрактов в нем.

5 Реализация

Для отслеживания различных операций в программе необходимо внедриться в ход ее выполнения и перехватывать обращения к методам, переменным и т. д. В Java это традиционно реализуется на уровне байт-кода. Это удобно, поскольку

он хорошо структурирован и вместе с тем достаточно просто организован. Кроме того, в отличие от исходного кода, байт-код классов доступен всегда.

Виртуальная Java машина (JVM) позволяет подключить к ней компоненту, реализующую особый интерфейс *java-агента*, который будет получать управление перед загрузкой очередного нового класса. Агенту передается массив байтов, содержащий байт-код загружаемого класса, который агент может *трансформировать*. В jDRD реализован этот интерфейс в отдельной компоненте. jDRD-агент анализирует байт-код загружаемых классов с помощью библиотеки ASM, находит интересующие инструкции (захват/освобождение монитора, обращение к разделяемой переменной, запуск потока и т. д.) и вставляет после них соответствующие внутренние вызовы детектора.

Таким образом, jDRD получает информацию об операциях синхронизации и обращениях к разделяемым данным в программе. Далее он обрабатывает их по алгоритму happens-before.

Остановимся подробней на обработке операций синхронизации. Когда jDRD встречает событие отправки синхронизационного сообщения, он должен увеличить собственную компоненту часов потока на единицу. После этого ему нужно сохранить свои часы так, чтобы они были доступны другому потоку, в котором произойдет соответствующее событие приема синхронизационного сообщения. Как отмечалось выше, с каждым синхронизационным событием можно связать уникальный синхронизационный объект. jDRD пользуется этим фактом и сохраняет часы потоков в хеш-таблице, ключами в которой являются те самые синхронизационные объекты. Для события освобождения монитора таким ключом будет ссылка на объект — владелец монитора, а для volatile-переменной — пара (*название переменной*, *ссылка на объект — владелец переменной*)[1].

Перейдем к отслеживанию синхронизационных контрактов. Их нужно трактовать как высокоуровневые операции синхронизации и ассоциировать с этими операциями искусственный синхронизационный объект. Поскольку описание контрактов содержится в отдельном xml-файле, они читаются и анализируются после запуска детектора. Далее для каждого контракта динамически генерируется класс, сущности которого впоследствии будут использоваться как синхронизационные объекты для данного контракта. Например, для контракта с рис. 1 будет создан следующий класс:

```
class CompositeKey1 {
    Object o1; //для примитивной связи владелец-владелец
    Object o2; //для примитивной связи параметр-параметр
}
```

[1] Для предотвращения утечек памяти в хеш-таблице используются не обычные (*сильные*, *strong reference*) ссылки на ключи, а слабые (*weak reference*). Ключевое свойство последних заключается в том, что если на объект остаются только слабые ссылки, то он может быть удален сборщиком мусора.

Когда jDRD встретит вызов метода ConcurrentMap.put, он обнаружит, что этот метод является частью синхронизационного контракта. В этот контракт входят две примитивные связи, каждая из которых опирается на Object. jDRD отыщет соответствующий класс-ключ (в данном случае это класс CompositeKey1) и создаст новый объект этого типа, который и будет синхронизационным объектом для данного исполнения контракта. Далее часы потока будут увеличены на единицу и записаны в хеш-таблицу. После этого jDRD перестает отслеживать операции синхронизации до того момента, когда метод put завершится. Чтобы распространить информацию о том, что отслеживание операций синхронизации в данном потоке стоит приостановить, в векторных часах потока есть переменная-флаг, которая выставляется в true, когда поток входит внутрь контрактного метода, и сбрасывается обратно в false при выходе из него. Если контрактный метод вызывается с уже установленным флагом, то флаг оставляется без изменений. Таким образом, сохраняется возможность корректной работы с контрактами различного уровня, часть из которых может быть использована при реализации других. При перехвате любой операции синхронизации jDRD в первую очередь проверяет состояние флага, и если оно равно true, он просто игнорирует эту операцию.

6 Преимущества и ограничения подхода

Алгоритм обнаружения гонок, применяемый в jDRD, базируется на учете всех отношений happens-before для выяснения корректности производимой операции доступа к данным. Поскольку отношения happens-before при выполнении операций синхронизации, заданные моделью памяти Java, «тотальны» (устанавливают отношение со всеми операциями, предшествующими точке синхронизации), любое событие синхронизации в коде программы имеет неограниченное воздействие на все отслеживаемые переменные и взаимодействующие потоки.

Задача детектора в идеальном случае — проверить корректность исполняемого кода только с учетом явно обозначенных контрактов используемых библиотек. Построение отношения happens-before с учетом всех произошедших операций синхронизации для кода, контракт которого не задает явных условий синхронизации, может привести к тому, что код, который с формальной точки зрения недостаточно синхронизирован, будет считаться корректным при рассмотрении всех промежуточных операций. Например, вызов стандартного java-метода System.out.err.print (используется для вывода сообщения об ошибке) содержит внутри себя критическую секцию. Если два потока вызывают этот метод поочередно, они синхронизируются между собой, но это является деталью реализации данного метода, побочным эффектом, а не декларированным свойством. Подобные синхронизационные события не только не представляют интереса, но и создают дополнительный «шум», затрудняющий обнаружение гонок.

Таким образом, производимые на основе описанных контрактов изъятия промежуточных синхронизационных событий позволяют не только уменьшить

общий объем работы анализатора, но и увеличить вероятность обнаружения гонок в некорректно синхронизированном коде, т. е. повышают точность метода.

Это подтверждается лабораторными экспериментами, которые проводились на трех приложениях:

(1) JTT — пользовательское клиентское приложение к системе отслеживания ошибок — порядка 400 классов, 10 потоков;

(2) QDTest — нагрузочный тест системы распространения котировок — 700 классов, 15 потоков;

(3) MARS — крупная многоцелевая мониторинговая система — 2000 классов, 30 потоков. Результаты приводятся отдельно для клиентской и серверной частей приложения.

jDRD запускался на Oracle JDK. Для обеспечения корректной обработки всех низкоуровневых средств синхронизации Java, основанных на атомарных инструкциях процессора, было добавлено отслеживание вызовов методов класса sun.misc.Unsafe[1].

Детектор jDRD запускался на перечисленных выше приложениях в двух режимах:

(1) *базовый режим* — отслеживание операций синхронизации во всем коде программы с использованием контрактов класса Unsafe;

(2) *juc-режим* — дополнительно описаны контракты для всех классов пакета java.util.concurrent, используемых приложениями.

В таблице представлены краткие результаты этих запусков.

Число обрабатываемых синхронизационных операций/контрактов в минуту и количество хранимых соответствующих векторных часов в различных режимах работы jDRD

Приложение	Режим	Количество синхронных операций в минуту	Количество синхронных часов	Количество контрактов в мин	Количество контрольных часов	Найдено гонок
JTT	Базовый	115 000	13 000	2 300	8 500	8
	Juc	28 000	7 000	600	750	10
QDTest	Базовый	15 000 000	6 100	209 000	1 400 000	1
	Juc	7 200 000	230	130 000	1 400 000	6
MARS client	Базовый	7 400 000	85 000	980 000	17 000	1
	Juc	4 300 000	72 000	730 000	24 000	5
MARS server	Базовый	1 650 000	15 000	360 000	5 500	2
	Juc	800 000	14 000	904 000	5 500	2

[1] В стандартной реализации JVM класс sun.misc.Unsafe содержит низкоуровневые примитивы (например, операции типа compare-and-swap). Все средства синхронизации Java базируются на synchronized/volatile/unsafe. Для других реализаций JVM нужно будет действовать иначе.

Использование контрактов снизило совокупное количество хранимых векторных часов и количество обрабатываемых операций синхронизации. Это, в свою очередь, снизило нагрузку на приложения и позволило в большинстве случаев обнаружить больше гонок. Все гонки, обнаруженные в базовом режиме, были также обнаружены и в juc-режиме, что свидетельствует о повышении точности поиска гонок.

С другой стороны, подход обладает рядом ограничений.

Во-первых, с помощью предложенных синхронизационных контрактов возможно описание лишь явно связанных методов. Можно представить себе и неявную связь, но это скорее исключение, чем правило. В пакете java.util.concurrent есть несколько таких методов, например метод newCondition объекта типа Lock. Этот метод возвращает объект типа Condition, внутренне связанный с исходным объектом типа Lock. В этом случае детектор просто «шагнет» внутрь этих классов и будет продолжать анализ.

Во-вторых, jDRD трактует методы, являющиеся частью синхронизационных контрактов, как атомарные, в то время как они таковыми не являются. Операция в программе, которая непосредственно обеспечивает синхронизацию потоков, обычно находится где-то внутри метода и может быть существенно отделена по времени от точек входа и выхода из него. Следовательно, на момент завершения работы метода информация может оказаться устаревшей, что может привести к ложным срабатываниям. Данная ситуация типична для любых неблокирующих средств синхронизации. В промежуток между самой операцией и ее обработкой поток может быть приостановлен, управление перейдет другим потокам и состояние системы изменится. Принудительное же заключение операции и ее обработки в атомарную секцию слишком сильно увеличит накладные расходы на программу. Сейчас идет работа над двухфазным механизмом обновления векторных часов при обработке подобных операций.

В-третьих, в рамках предложенного подхода методы исключенных объектов рассматриваются как операции лишь над данными, содержащимися в их объекте-владельце. Разумеется, это не всегда так и довольно часто методы модифицируют объекты, переданные им как параметры. Данное обстоятельство служит текущим техническим ограничением реализации подхода и является предметом для дальнейшей доработки детектора. Передача объекта в метод должна рассматриваться как обращение к этому объекту как минимум на чтение, а возможно, и на запись, и нужно иметь возможность это конфигурировать.

Наконец, предложенный язык описания контрактов позволяет описывать только внешние, декларированные точки «входа» в контракт и «выхода» из него. Однако описанный контрактами и исключенный из анализа объект может сам вызывать методы переданных ему параметров. Например, ConcurrentHashMap вызывает у переданных ей объектов-ключей метод equals, а класс Executor для ведения внутреннего журнала может обратиться к переданной ему задаче за текстовым представлением, вызвав ее метод toString. Описание подобных ситуаций и их корректная динамическая обработка так-

же представляют собой одно из важных направлений дальнейшего исследования.

7 Заключение

Одной из принципиальных проблем динамического анализа программ является обеспечение сочетания точности и глубины анализа и приемлемого уровня накладных расходов. В задаче автоматического поиска гонок эта проблема особенно актуальна, поскольку количество операций синхронизации и обращений к разделяемым данным очень велико. Если область программы, в которой необходимо отслеживать обращения к разделяемым переменным является скорее вопросом конфигурации и выделения наиболее «интересных» и опасных участков кода, то операции синхронизации нужно отслеживать во всем коде программы, чтобы не пропустить информацию о синхронизации потоков.

В данной работе предлагается концепция синхронизационных контрактов, в основе которой лежит идея отслеживания не всех операций синхронизации, происходящих в программе, а лишь явно декларированных. Вводится понятие синхронизационного контракта как пары явно связанных методов и предлагается язык для их описания, основанный на xml-нотации. Поддержка и корректная обработка описанных таким образом контрактов была реализована в детекторе jDRD, который посредством трансформирования байт-кода внедряется в java-программу, отслеживает в ней важные события (в том числе и методы, описанные в контрактах) и обрабатывает их по алгоритму happens-before. Лабораторное тестирование показывает эффективность использования контрактов — сокращается как число обрабатываемых операций синхронизации, так и количество векторных часов, которые необходимо хранить для отслеживания отношения happens-before.

Подход обладает рядом ограничений, над устранением которых планируется продолжить работу. Также проводится внедрение jDRD в процесс разработки программного обеспечения и его промышленная апробация, результаты которой планируется опубликовать в дальнейшем.

Приложение

Язык описания синхронизационных контрактов

Для описания happens-before контрактов пар методов различных классов предназначен тег `Syncs`, содержащий несколько тегов `Sync` — по одному на каждый контракт. В описании контракта указываются все примитивные связи, образующие связь между методами (тег `Links` содержит по одному тегу на каждую примитивную связь), и описание вызовов методов, являющихся отправкой и приемкой отношения happens-before (теги `Send` и `Receive` соответственно):

```
<!ELEMENT Syncs ( Sync+ ) >
<!ELEMENT Sync ( Links, Send, Receive ) >
<!ELEMENT Receive ( MethodCall ) >
<!ELEMENT Send ( MethodCall ) >
<!ELEMENT Links ( Link+ ) >
```

Вызов метода описывается в теге `MethodCall`: указывается класс-владелец метода, название метода и его дескриптор во внутренней нотации JVM.

Примитивная связь описывается тегом `Link`:

```
<!ELEMENT Link EMPTY >
<!ATTLIST Link
    receive (owner|param) #REQUIRED >
    receive-number CDATA #IMPLIED >
    send (owner|param) #REQUIRED >
    send-number CDATA #IMPLIED >
```

Атрибуты `send` и `send-number` соответствуют левой части примитивной связи, а `receive` и `receive-number` — правой. Обе они могут быть либо типа «владелец», либо типа «параметр». Если правая часть типа «владелец», то `receive` имеет значение «owner», а `receive-number` не указывается. В другом случае `receive` имеет значение «param», а `receive-number` содержит номер параметра в сигнатуре метода (нумерация начинается с нуля). Аналогично для атрибутов `send` и `send-number`. Пример такого контракта приведен на рис. 1.

Если нужно описать контракты нескольких пар методов одного и того же класса, это можно сделать с помощью тега `Multiple-Syncs`, состоящего из нескольких тегов `Multiple-Sync`, каждый из которых соответствует одному классу. Полное имя класса указывается в атрибуте owner, а связь между вызовами методов описывается в теге `Multiple-Links`:

```
<!ATTLIST Multiple-Sync owner ID #REQUIRED >
<!ELEMENT Multiple-Syncs ( Multiple-Sync+ ) >
<!ELEMENT Multiple-Sync ( Multiple-Links, Call+ ) >
<!ELEMENT Multiple-Links ( Multiple-Link+ ) >
```

Пример такого контракта приведен на рис. 2.

```
<Multiple-Sync owner="java.util.concurrent.atomic.AtomicBoolean">
    <Multiple-Links>
        <Multiple-Link type="owner"/>
     </Multiple-Links>
     <Call type="receive" name="get" descriptor="()Z"/>
     <Call type="full" name="compareAndSet" descriptor="(ZZ)Z"
           shouldReturnTrue="true"/>
     <Call type="send" name="set" descriptor="(Z)V"/>
     <Call type="full" name="getAndSet" descriptor="(Z)Z"/>
</Multiple-Sync>
```

Рис. 2 Пример описания синхронизационных контрактов для нескольких методов одного класса

Литература

1. *Netzer R., Miller B.* What are race conditions? Some issues and formalizations // ACM Lett. Programming Languages Syst., 1992. Vol. 1. No. 1. P. 74–88.
2. *Leino K., Nelson G., Saxe J.* ESC/Java user's manual: SRC technical note 2000-002. http://www.hpl.hp.com/techreports/Compaq-DEC/SRC-TN-2000-002.pdf.
3. *Engler D., Ashcraft K.* RacerX: Effective, static detection of race conditions and deadlocks // 19th ACM Symposium on Operating Systems Principles Proceedings. — New York, NY, USA: ACM, 2003. P. 237–252.
4. *Naik M., Aiken A., Whaley J.* Effective static race detection for Java // 2006 ACM SIGPLAN Conference on Programming Language Design and Implementation Proceedings. — New York, NY, USA: ACM, 2006. P. 308–319.
5. *Savage S., Burrows M., Nelson G., Sobalvarro P., Anderson T.* Eraser: A dynamic data race detector for multithreaded programs // ACM Trans. Comput. Syst., 1997. Vol. 15. Issue 4. P. 391–411.
6. *Christiaens M., Brosschere K.* TRaDe: A topological approach to on-the-fly race detection in Java programs // 2001 Symposium on JavaTM Virtual Machine Research and Technology Symposium Proceedings. — Berkeley, CA, USA: USENIX Association, 2001. Vol. 1. P. 105–116.
7. *Choi J., Lee K., Loginov A., O'Callahan R., Sarkar V., Sridharan M.* Efficient and precise data-race detection for multithreaded object-oriented programs // ACM SIGPLAN 2002 Conference on Programming Language Design and Implementation Proceedings. — New York, NY, USA: ACM, 2002. P. 258–269.
8. *Pozniansky E., Schuster A.* Efficient on-the-fly data race detection in multithreaded C++ programs // 9th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming Proceedings. — New York, NY, USA: ACM, 2003. P. 179–190.
9. *Elmas T., Qadeer S., Tasiran S.* Goldilocks: A race and transaction-aware Java runtime // 2007 ACM SIGPLAN Conference on Programming Language Design and Implementation Proceedings. — New York, NY, USA: ACM, 2002. P. 245–255.
10. *Flanagan C., Freund S. N.* FastTrack: Efficient and precise dynamic race detection // 2009 ACM SIGPLAN Conference on Programming Language Design and Implementation Proceedings. — New York, NY, USA: ACM, 2009. P. 121–133.
11. *Qi Y., Das R., Luo Z., Trotter M.* MulticoreSDK: A practical and efficient data race detector for real-world applications // 4th IEEE Conference (International) on Software Testing, Verification, and Validation (ICST 2011) Proceedings. — Los Alamitos, CA, USA: IEEE, 2011. P. 309–318.
12. ThreadSanitizer for Java: A run-time detector of data races. http://code.google.com/p/java-thread-sanitizer.
13. *Lamport L.* Time, clocks and the ordering of events in a distributed system // Commun. ACM, 1978. Vol. 21. No. 7. P. 558–565.
14. Java language specification. 3rd ed. Ch. 17. Threads and locks. 17.4.5. Happens-before Order. http://docs.oracle.com/javase/specs/jls/se7/html/jls-17.html#jls-17.4.5.
15. *Mattern F.* Virtual time and global states of distributed systems // Workshop (International) on Parallel and Distributed Algorithms Proceedings / Eds. M. Corsnard, P. Quinton, M. Raynal, Y. Robert. — Amsterdam: Elsevier, 1989. P. 215–226.

16. *Трифанов В. Ю., Цителов Д. И.* Динамические средства обнаружения гонок в параллельных программах // Компьютерные инструменты в образовании, 2011. № 5. С. 3–15.
17. Трифанов В. Ю., Цителов Д. И. Статические и post-mortem средства обнаружения гонок в параллельных программах // Компьютерные инструменты в образовании, 2011. № 6. С. 3–13.
18. Package java.util.concurrent: Utility classes commonly useful in concurrent programming. http://download.oracle.com/javase/6/docs/api/java/util/concurrent/package-summary.html.

Поступила в редакцию 20.01.14

DATA RACE DETECTION IN JAVA PROGRAMS USING SYNCHRONIZATION CONTRACTS

***D. Tsitelov*[1] *and V. Trifanov*[2]**

[1]Expert-Sistema Ltd., 10/1 Barochnaya Str., St. Petersburg 197022, Russian Federation
[2]Expert-Sistema SZ Ltd., 10/1 Barochnaya Str., St. Petersburg 197022, Russian Federation

Abstract: Data race occurs in a multithreaded program when several threads simultaneously access the same memory location and at least one of them has a write access. Data races are hardly reproducible and can damage global data structures; so, research in the area of automatic race detection methods has been carried out for more than 20 years. This article focuses on the issue of improving performance of dynamic race detection in Java programs without loss of precision. Synchronization contracts — partial specifications of multithreaded behavior — are introduced for solving this problem. Using contracts allows excluding parts of application's code that are not interesting from the race detection perspective (e. g., external libraries). The paper also covers advantages and restrictions of the approach, the contracts specification language, and some implementation details.

Keywords: multithreading; data race; dynamic analysis; automatic error detection

DOI: 10.14357/08696527140208

References

1. Netzer, R., and B. Miller. 1992. What are race conditions? Some issues and formalizations. *ACM Lett. Programming Languages Syst.* 1(1):74–88.
2. Leino, K., G. Nelson, and J. Saxe. 2001. ESC/Java user's manual: SRC Technical note 2000-002. Available at: http://www.hpl.hp.com/techreports/Compaq-DEC/SRC-TN-2000-002.pdf (accessed March 28, 2014).

3. Engler, D., and K. Ashcraft. 2003. RacerX: Effective, static detection of race conditions and deadlocks. *19th ACM Symposium on Operating Systems Principles Proceedings*. New York, NY, USA: ACM. 237–252.
4. Naik, M., A. Aiken, and J. Whaley. 2006. Effective static race detection for Java. *2006 ACM SIGPLAN Conference on Programming Language Design and Implementation Proceedings*. New York, NY, USA: ACM. 308–319.
5. Savage, S., M. Burrows, G. Nelson, P. Sobalvarro, and T. Anderson. 1997. Eraser: A dynamic data race detector for multithreaded programs. *ACM Trans. Comput. Syst.* 15(4):391–411.
6. Christiaens, M., and K. Brosschere. 2001. TRaDe: A topological approach to on-the-fly race detection in Java programs. *2001 Symposium on Java Virtual Machine Research and Technology Symposium Proceedings*. Berkley, CA, USA: USENIX Association. 1:105–116.
7. Choi, J., K. Lee, A. Loginov, R. O'Callahan, V. Sarkar, and M. Sridharan. 2002. Efficient and precise data-race detection for multithreaded object-oriented programs. *ACM SIGPLAN 2002 Conference on Programming Language Design and Implementation Proceedings*. New York, NY, USA: ACM. 258–269.
8. Pozniansky, E., and A. Schuster. 2003. Efficient on-the-fly data race detection in multithreaded C++ programs. *9th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming Proceedings*. New York, NY, USA: ACM. 179–190.
9. Elmas, T., S. Qadeer, and S. Tasiran. 2007. Goldilocks: A race and transaction-aware Java runtime. *2007 ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI'07) Proceedings*. New York, NY, USA: ACM. 245–255.
10. Flanagan, C., and S. Freund. 2009. FastTrack: Efficient and precise dynamic race detection. *ACM Conference on Programming Language Design and Implementation*. New York, NY, USA: ACM. 121–133.
11. Qi, Y., R. Das, Z. Luo, and M. Trotter. 2011. MulticoreSDK: A practical and efficient data race detector for real-world applications. *Software Testing, Verification and Validation (ICST) Proceedings*. Los Alamitos, CA, USA: IEEE. 309–318.
12. ThreadSanitizer for Java: A run-time detector of data races. Available at: http://code.google.com/p/java-thread-sanitizer/ (accessed March 28, 2014).
13. Lamport, L. 1978. Time, clocks and the ordering of events in a distributed system. *Commun. ACM* 21(7):558–565.
14. Java language specification. 3rd ed. Threads and locks. Happens-before order. Available at: http://docs.oracle.com/javase/specs/jls/se7/html/jls-17.html#jls-17.4.5 (accessed March 28, 2014).
15. Mattern, F. 1989. Virtual time and global states of distributed systems. *Workshop on Parallel and Distributed Algorithms Proceedings*. Eds. M. Cosnard, P. Quinton, M. Raynal, and Y. Robert. Amsterdam: Elsevier. 215–226.
16. Trifanov, V. Yu., and D. I. Tsitelov. 2011. Dinamicheskie sredstva obnaruzheniya gonok v parallel'nykh programmakh [Dynamic data race detectors for parallel programs]. *Komp'yuternye Instrumenty v Obrazovanii* [Computer Instruments in Education] 5:3–15.
17. Trifanov, V. Yu., and D. I. Tsitelov. 2011. Staticheskie i post-mortem sredstva obnaruzheniya gonok v parallel'nykh programmakh [Static and post-mortem data race detectors for parallel programs]. *Komp'yuternye Instrumenty v Obrazovanii* [Computer Instruments in Education] 6:3–13.

18. Package java.util.concurrent: Utility classes commonly useful in concurrent programming. Documentation of java.util.concurrent package. Available at: http://download.oracle. com/javase/6/docs/api/java/util/concurrent/package-summary.html (accessed March 28, 2914).

Received January 20, 2014

Contributors

Tsitelov Dmitry I. (b. 1974) — Project Lead, Expert-Sistema Ltd., 10/1 Barochnaya Str., St. Petersburg 197022, Russian Federation; tsitelov@acm.org

Trifanov Vitaly Yu. (b. 1988) — Candidate of Science (PhD) in technology, senior software developer, Expert-Sistema SZ Ltd. 10/1 Barochnaya Str., St. Petersburg 197022, Russian Federation; vitaly.trifanov@gmail.com

МЕТОДИКА ИЗВЛЕЧЕНИЯ ПОСЛОВНЫХ ПЕРЕВОДНЫХ СООТВЕТСТВИЙ ИЗ ПАРАЛЛЕЛЬНЫХ ТЕКСТОВ С ПРИМЕНЕНИЕМ МОДЕЛЕЙ ДИСТРИБУТИВНОЙ СЕМАНТИКИ

***Ю. И. Морозова*[1], *Е. Б. Козеренко*[2], *М. М. Шарнин*[3]**

Аннотация: Данная работа посвящена актуальным проблемам исследования лингвистических единиц с использованием корпусных методов. В работе дается определение задачи извлечения переводных соответствий из параллельных текстов, рассматриваются существующие подходы к решению данной задачи и предлагается подход, основанный на применении моделей дистрибутивной семантики. Приводится описание разработанной авторами теоретической модели извлечения переводных соответствий, а также ее компьютерной реализации. Для целей исследования был создан тестовый параллельный корпус, содержащий тексты патентов на русском и французском языках. Приводятся результаты применения дистрибутивно-семантической методики, полученные в рамках эксперимента по извлечению переводных соответствий из тестового корпуса.

Keywords: выравнивание параллельных текстов; параллельный корпус; дистрибутивная семантика; модель векторных пространств

DOI: 10.14357/08696527140209

1 Введение

В течение последних двух десятилетий появилось большое количество текстовых массивов на различных языках в машиночитаемом формате. Текстовые корпусы являются важным источником информации о функционировании языковой системы. Методы компьютерной лингвистики позволяют автоматически извлекать информацию о морфологических, синтаксических и семантических свойствах языковых единиц из текстовых корпусов. Данная информация используется в системах автоматической обработки текстов, повышая качество работы систем.

Одной из разновидностей текстовых корпусов являются корпусы параллельных текстов, содержащие тексты на нескольких языках, являющиеся взаимными

[1]Институт проблем информатики Российской академии наук, yulia-ipi@yandex.ru
[2]Институт проблем информатики Российской академии наук, kozerenko@mail.ru
[3]Институт проблем информатики Российской академии наук, keywen1@mail.ru

переводами. Подобные корпусы являются ценным ресурсом для создания систем машинного перевода, так как они содержат в себе информацию о правилах перевода как на уровне отдельных слов и словосочетаний, так и на уровне синтаксических конструкций.

Задача извлечения переводных соответствий из параллельных текстов является актуальной для создания систем машинного перевода в связи с необходимостью автоматического пополнения переводных словарей и их настройки на различные предметные области. Данная задача является частью более общей задачи выравнивания параллельных текстов, которая заключается в нахождении фрагментов параллельных текстов, соответствующих друг другу: документов, абзацев, предложений, словосочетаний и слов.

Пословное выравнивание параллельных текстов понимается как объект, указывающий соответствующие друг другу слова в параллельном тексте [1]. Во многих случаях существует несколько равноправных вариантов установления соответствий (например, при переводе идиоматических конструкций), поэтому понятие наиболее правильного выравнивания является субъективным.

Приведем математическое определение пословного выравнивания. Пусть даны два предложения, являющиеся взаимными переводами. Предложения рассматриваются как две последовательности слов: одна на исходном языке, обозначаемая как $s_1^J = s_1, \ldots, s_J$ (от *англ.* source language), а другая — на целевом языке, обозначаемая как $t_1^I = t_1, \ldots, t_I$ (от *англ.* target language). Под выравниванием a между двумя предложениями понимается отображение множества позиций слов исходного предложения $\{1, \ldots, J\}$ во множество позиций слов целевого предложения $\{1, \ldots, I\}$:

$$a : \; j \to i \, .$$

Выравнивание обычно изображается с помощью чисел — идентификаторов слов, из которых состоят параллельные предложения. Приведем пример, полученный при использовании программы для пословного выравнивания GIZA++ [2].

Пара параллельных предложений с идентификаторами слов:

Предложение на исходном языке:

Штифт (1) *для* (2) *использования* (3) *в* (4) *стоматологии* (5).

Перевод на целевой язык:

NULL (0) *Implant* (1) *à* (2) *usage* (3) *dentaire* (4).

Полученное выравнивание:

$a : \; \{1 \to 1, 2 \to 0, 3 \to 3, 4 \to 0, 5 \to 4\}$.

Из данного выравнивания следует, что слова русского предложения с номерами 2 и 4 (*для*, *в*) не имеют соответствий во французском языке (они соответствуют пустому слову NULL). Для остальных слов устанавливаются следующие соответствия: *штифт* → *implant*, *использования* → *usage*, *стоматологии* → *dentaire*.

Выравнивание также может быть представлено в виде рисунка, что позволяет быстро оценить степень сходства порядка слов в двух языках. Для приведенной пары выравнивание отражено на рис. 1.

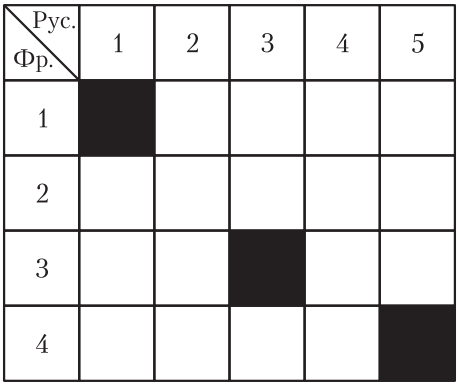


Рис. 1 Пословное выравнивание между двумя предложениями

В соответствии с определением выравнивания каждое слово исходного языка формально должно иметь соответствие в виде одного слова целевого языка, даже если содержательно это слово в данной паре предложений остается без перевода. Для выполнения этого требования к словам целевого языка добавляется специальное «пустое» слово. Слова целевого языка могут не иметь соответствий в исходном языке (в данном примере французское слово à не имеет соответствий в русском языке) или же соответствовать нескольким словам (в данном примере пустое слово NULL соответствует нескольким словам, однако и непустые слова целевого языка тоже могут соответствовать нескольким словам исходного языка). Иными словами, отношения между двумя языками в данной модели выравнивания не являются симметричными.

2 Модели выравнивания параллельных текстов

Существует два основных типа статистических моделей пословного выравнивания параллельных текстов: эвристические и самообучающиеся модели. Во многих работах также предлагается дополнительно использовать лингвистические фильтры, чтобы отбрасывать неправильные переводные соответствия, полученные в результате работы статистических алгоритмов.

Эвристические модели используют функцию подобия (ассоциативные меры) между словами двух языков. Чаще всего используются различные варианты формулы Дайса:

$$\mathrm{dice}\,(i,j) = \frac{2C(t_i, s_j)}{C(t_i)C(s_j)}\,,$$

где s_j — слово, стоящее в предложении исходного языка на позиции j; t_i — слово, стоящее в предложении целевого языка на позиции i; $C(t_i, s_j)$ — частота совместной встречаемости данных слов в соответствующих друг другу фрагментах параллельного корпуса; $C(s_j)$ — частота встречаемости слова s_j в текстах на исходном языке; $C(t_i)$ — частота встречаемости слова t_i в текстах на целевом языке.

Для каждой пары соответствующих друг другу предложений строится матрица со значениями ассоциативных мер между каждым словом исходного предложения и каждым словом целевого предложения. В качестве выравнивания $a_j = i$ для позиции j выбирается такое слово, у которого мера ассоциации является наибольшей:

$$a_j = \arg \max_i \{\mathrm{dice}\,(i, j)\}\,.$$

Эвристические модели просты для понимания и реализации, поэтому они широко применяются для пословного выравнивания. Недостатком данных моделей является произвольно заданная функция подобия. Произвольность делает эвристические модели менее состоятельными, чем модели, основанные на обучении.

В рамках подхода, основанного на обучении, моделируется вероятность перевода $\Pr\left(t_1^I|s_1^J\right)$, описывающая отношения между предложением на исходном языке s_1^J и предложением на целевом языке t_1^I. Модель перевода представляется в виде сочетания модели перевода и модели языка с использованием правила Байеса. Вероятность перевода предложения исходного языка s_1^J предложением целевого языка t_1^I представляется в следующем виде:

$$\arg \max_{t_1^I} \Pr\left(t_1^I|s_1^J\right) = \arg \max_{t_1^I} \Pr\left(s_1^J|t_1^I\right) \cdot \Pr(t_1^I)\,,$$

где $\Pr\left(t_1^I|s_1^J\right)$ — вероятность того, что предложение t_1^I является переводом предложения s_1^J; $\Pr(s_1^J|t_1^I)$ — вероятность перевода в обратном порядке (в соответствии с правилом Байеса); $\Pr(t_1^I)$ — вероятность предложения t_1^I. В самообучающихся моделях используются различные распределения вероятностей, которые определяют вероятность перевода $\Pr(s_1^J|t_1^I)$ и вероятность предложения t_1^I. Распределения вероятностей автоматически определяются на этапе обучения системы по корпусу текстов. Параметры оптимизируются с учетом критерия максимального правдоподобия, для чего обычно используется алгоритм максимизации ожидания.

Наиболее известными моделями выравнивания параллельных текстов, основанными на обучении, являются скрытая марковская модель выравнивания, описанная в [3], и модели IBM №№ 1–5, приведенные в [1].

Использование лингвистических знаний позволяет отбрасывать неправильные переводные соответствия, полученные в результате работы статистических моделей, и улучшать точность работы системы [4]. Например, в работе [5] предлагается метод выравнивания по словам, относящимся к значимым частям речи, при этом служебные слова не учитываются. В работе [6] предлагается дополнять статистические методы выравнивания лингвистическим сопоставлением слов на следующих уровнях: орфографическом (например, *organization* в английском языке и *organización* в испанском языке); лексическом (при наличии переводных словарей для изучаемых языков); морфологическом (совпадение частей речи);

синтаксическом (совпадение синтаксических функций); семантическом (совпадение семантических классов).

В работе [2] описан эксперимент по сравнению различных статистических моделей выравнивания. Результаты эксперимента свидетельствуют о том, что самообучающиеся модели дают более точные результаты, чем эвристические модели.

3 Модели дистрибутивной семантики

Дистрибутивный анализ — это метод исследования языка, основанный на изучении окружения (дистрибуции, распределения) отдельных единиц в тексте и не использующий сведений о полном лексическом или грамматическом значении этих единиц [7]. Дистрибутивный анализ был впервые предложен Л. Блумфилдом в 1920-х гг. и применялся, главным образом, в фонологии и морфологии. В ходе исследований выяснилось, что, используя контексты в качестве исходных данных, можно выделить основные единицы языка (фонемы, морфемы, слова, словосочетания), объединить их в классы и установить отношения сочетаемости между этими классами.

В течение последних двух десятилетий было предпринято множество успешных попыток применения метода дистрибутивного анализа к изучению семантики. Была разработана дистрибутивно-семантическая методика, позволяющая в автоматическом режиме сопоставлять контексты, в которых встречаются различные слова, и устанавливать семантические связи между ними.

В качестве вычислительного инструмента дистрибутивно-семантической методики используется линейная алгебра. Информация о дистрибуции лингвистических единиц представляется в виде многоразрядных векторов, которые образуют словесное пространство. Векторы соответствуют лингвистическим единицам (словам или словосочетаниям), а измерения векторов соответствуют контекстам. Размер контекста определяется целями исследования. Для установления синтагматических связей используются контексты, ограниченные одним или двумя соседними словами, для установления парадигматических связей используется контекстное «окно» размером 5–10 слов, для установления тематических связей используются контексты размером 50 слов и больше. Векторы состоят из чисел, которые показывают, сколько раз данное слово или словосочетание встретилось в данном контексте в данном корпусе текстов. На рис. 2 в схематическом виде представлен пример словесного пространства, которое опи-

$$A_{m,n} = \begin{array}{c} \\ \textbf{coffee} \\ w_2 \\ \textbf{tea} \\ \vdots \\ w_m \end{array} \begin{array}{c} \begin{array}{ccccc} w_1 & \textbf{drink} & w_3 & \cdots & w_m \end{array} \\ \begin{bmatrix} 0 & \mathbf{1} & 0 & \cdots & 1 \\ 1 & 0 & 2 & \cdots & 0 \\ 0 & \mathbf{2} & 0 & \cdots & 3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 1 & \cdots & 0 \end{bmatrix} \end{array}$$

Рис. 2 Словесное пространство для слов английского языка coffee и tea

сывает дистрибутивные характеристики лексем английского языка (в качестве контекста используется соседнее слово слева).

Семантическая близость между лингвистическими единицами вычисляется как расстояние между векторами. Наиболее популярной мерой близости между векторами в лингвистических исследованиях является косинусная мера, вычисляемая по формуле

$$\text{sim}\,(\vec{x},\vec{y})\,\frac{xy}{|x|\cdot|y|}=\frac{\sum\limits_{i=1}^{n}x_iy_i}{\sqrt{\sum\limits_{i=1}^{n}x_i^2\cdot\sum\limits_{i=1}^{n}y_i^2}},$$

где x, y — обозначения двух векторов; x_i, y_i — значения элементов словесного пространства, соответствующих i-му измерению векторов.

Модели дистрибутивной семантики находят широкое применение в исследованиях, связанных с семантическими моделями естественного языка, и имеют разнообразный спектр потенциальных и действующих приложений. Дистрибутивные модели были успешно применены для решения следующих задач автоматической обработки текста:

- выявление семантической близости слов и словосочетаний;
- автоматическая кластеризация слов по степени их семантической близости;
- автоматическая генерация тезаурусов и двуязычных словарей;
- разрешение лексической неоднозначности;
- расширение запросов за счет ассоциативных связей;
- определение тематики документа;
- кластеризация документов для информационного поиска;
- извлечение знаний из текстов;
- построение семантических карт различных предметных областей;
- моделирование перифраз;
- определение тональности высказывания;
- моделирование сочетаемостных ограничений глагола.

Векторные модели были успешно применены для решения задачи извлечения переводных эквивалентов из параллельных корпусов текстов. В работе [8] методика дистрибутивной семантики использовалась для извлечения однословных переводных соответствий из двух корпусов параллельных текстов законодательных актов: шведско-испанского и англо-немецкого, выровненных на уровне документов. Тексты были подвергнуты лемматизации, затем леммы из обоих

языков были помещены в единое словесное пространство, в котором в качестве контекстов использовались идентификационные номера документов. В качестве перевода для слова исходного языка выбиралось слово целевого языка, имеющее наиболее близкий вектор в соответствии с косинусной мерой. Переводы, полученные в результате работы программы, были сравнены с существующими переводными словарями: если перевод был найден в словаре, то переводное соответствие засчитывалось как верное (даже в том случае если слово являлось частью составного слова или частью устойчивого словосочетания). Для оценки результатов использовалась мера точности (precision), вычисляемая по следующей формуле:

$$\text{precision} = \frac{C}{S},$$

где C — количество правильных переводных соответствий в полученном словаре; S — размер словаря.

Средний показатель уровня точности при сопоставлении с существующими словарями составил 60%. Для проверки этого показателя экспертов-переводчиков попросили просмотреть переводные соответствия, которые не прошли проверку на соответствие «золотому стандарту» (существующему переводному словарю). Выяснилось, что 30% отброшенных переводных соответствий представляют собой правильные переводы. Таким образом, при ручной оценке результатов точность выделения переводных соответствий составила 72%.

В работе [6] для извлечения однословных переводных соответствий использовались три разных метода: алгоритм K-Vec с булевыми частотами, алгоритм K-Vec с абсолютными частотами и модель IBM № 2. Вначале тексты исходного и целевого языков были разделены на сегменты произвольного размера от параграфа до нескольких слов. Затем слова были помещены в векторное пространство, в котором в качестве контекстов выступали текстовые сегменты. Для каждого слова был сформирован список возможных переводов, из которого был выбран наилучший перевод с помощью мер ассоциации: Pointwise Mutual Information, T-score, Log-likelihood ratio, Dice coefficient. Затем эта процедура была повторена в противоположном направлении перевода. Конечным результатом стал усредненный результат обоих экспериментов. В данной работе также было применено сопоставление слов на уровне букв, позволяющее находить переводные соответствия родственных слов и имен собственных. В качестве материала были выбраны тексты художественных произведений на испанском и английском языках, выровненные на уровне предложений. По результатам ручной оценки точность выделения переводных соответствий составила 53%.

4 Эксперимент по применению моделей дистрибутивной семантики для извлечения переводных соответствий из параллельных текстов

Цель данной работы заключалась в разработке дистрибутивно-семантической модели и ее проверке на тестовом примере. В ходе исследования был создан

корпус параллельных текстов научных патентов на русском и французском языках. Тексты патентов были получены с сайта Европейского патентного агентства [9]. В процессе создания исследовательского корпуса были пройдены следующие этапы:

1. Разделение текстов на предложения с помощью программы, являющейся частью системы морфологического анализа текстов, разработанной в лаборатории компьютерной лингвистики ИПИ РАН [10].
2. Выравнивание текстов на уровне предложений с помощью программы Bilingual Sentence Aligner [11] с последующим ручным редактированием в визуальном редакторе Make Bilingua [12].
3. Загрузка текстов в онлайн-систему ведения корпусов Sketch Engine [13].

При загрузке текстов в систему Sketch Engine автоматически производится морфологическая разметка: словоформы снабжаются информацией о начальной форме слова, части речи и грамматических характеристиках. Модуль морфологического анализа системы Sketch Engine был разработан в рамках проекта по созданию статистической системы морфологического анализа Tree Tagger, основанной на деревьях решений [14]. Модуль морфологического анализа для русского языка был разработан Шаровым и Нивре [15].

Для решения задачи извлечения переводных соответствий использовались методы дистрибутивной семантики. Была построена модель семантического векторного пространства со следующими параметрами:

– тип изучаемых единиц: лексемы;
– тип контекста: пары выровненных предложений;
– количественная оценка частоты встречаемости изучаемой единицы в данном контексте: булева частота (1 — если слово встретилось в данном контексте, 0 — в противном случае);
– метод вычисления расстояния между векторами: косинусная мера.

Компьютерная реализация модели семантического векторного пространства была выполнена М. М. Шарниным с коллегами [16]. Алгоритм работы программы состоит из следующих этапов:

1. Из всех слов выделяется 1000 самых частых слов для обоих языков.
2. Для выделенных слов строится словесное пространство.
3. Вычисляется мера близости между векторами данного пространства (с использованием косинусной меры).
4. В качестве переводного соответствия для каждого слова русского языка выбирается наиболее близкое к нему слово французского языка.

В результате применения модели семантического векторного пространства был получен список однословных переводных соответствий, например:

Сравнение данной работы с работами других авторов

Источник	Жанр	Контекст	Лингвистический фильтр	Точность
Данная работа	Деловая проза	Предложение	Да	89%
[6]	Художественный текст	Предложение	Да	53%
[8]	Деловая проза	Документ	Нет	73%

moyen → *средство*, *exemple* → *например*, *caract ériser* → *отличать*. Затем к данному списку был применен лингвистический фильтр: слова, принадлежащие к разным частям речи, были исключены из списка. Была произведена ручная оценка получившегося списка, в результате которой выяснилось, что точность извлечения однословных переводных соответствий составляет 89%.

В таблице приводится сравнение результатов, полученных в данной работе, с результатами двух работ по извлечению однословных переводных соответствий, описанных в разд. 3.

Более высокий показатель точности, полученный авторами статьи, может объясняться тем, что в эксперименте использовался корпус параллельных текстов высокого качества: тексты были выровнены на уровне предложений, а не документов, выравнивание было проверено вручную. Применение лингвистического фильтра на совпадение частей речи позволило отбросить множество неверных соответствий. Использование текстов деловой прозы, а не художественных текстов могло также внести свой вклад в повышение точности выделения переводных соответствий.

5 Заключение

Извлечение переводных соответствий из параллельных текстов — актуальная задача, которая возникает при проектировании систем машинного перевода, так как переводные словари нужно пополнять и настраивать на различные предметные области. В работе дается обзор существующих подходов к решению данной задачи, при этом наибольшее внимание уделяется дистрибутивно-семантическому подходу. Авторами была разработана дистрибутивно-семантическая модель извлечения переводных соответствий из параллельных текстов. Эксперименты показали, что при использовании данной модели точность извлечения переводных соответствий составляет 89%, что превосходит результаты, полученные другими авторами. Дальнейшие исследования будут направлены на извлечение переводных соответствий на уровне словосочетаний.

Литература

1. *Brown P. F., Della Pietra S. A., Della Pietra V. J., Mercer R. L.* The mathematics of statistical machine translation: Parameter estimation // Comput. Linguistics, 1993. Vol. 19. No. 2. P. 263–311.

2. *Och F. J., Ney H.* A systematic comparison of various statistical alignment models // Comput. Linguistics, 2003. Vol. 29. No. 1. P. 19–51.
3. *Vogel S., Ney H., Tillmann Ch.* HMM-based word alignment in statistical translation // COLING'96: 16th Conference on Computational Linguistics Proceedings. — Stroudsburg, PA, USA: Association for Computational Linguistics, 1996. Vol. 2. P. 836–841.
4. *Козеренко Е. Б.* Лингвистические фильтры в статистических моделях машинного перевода // Информатика и её применения, 2010. Т. 4. Вып. 2. С. 83–92.
5. *Masahiko H., Yamazaki T.* High-performance bilingual text alignment using statistical and dictionary information // ACL'96: 34th Annual Meeting on Association for Computational Linguistics Proceedings. — Stroudsburg, PA, USA: Association for Computational Linguistics, 1996. P. 131–138.
6. *Cendejas E., Barcelo G., Gelbukh A., Sidorov G.* Incorporating linguistic information to statistical word-level alignment // 14th Iberoamerican Conference on Pattern Recognition Proceedings. — Berlin, Germany: Springer, 2009. P. 387–394.
7. Лингвистический энциклопедический словарь / Гл. ред. В. Н. Ярцева. — М.: Советская энциклопедия, 1990.
8. *Sahlgren M., Karlgren J.* Automatic bilingual lexicon acquisition using random indexing of parallel corpora // J. Natural Language Eng. (Special Issue on Parallel Texts), 2005. Vol. 11. Issue 3. P. 327–341.
9. European patent office. http://www.epo.org/index.html (30.03.14).
10. *Сомин Н. В., Кузнецов И. П., Николаев В. Г., Соловьева Н. С., Мацкевич А. Г.* Методы устранения неопределенностей блока лексико-морфологического анализа при извлечении знаний из текстов естественного языка // Системы и средства информатики, 2011. Т. 21. Вып. 2. С. 97–115.
11. Bilingual Sentence Aligner. http://research.microsoft.com/en-us/downloads/aafd5dcf-4dcc-49b2-8a22-f7055113e656 (30.03.14).
12. Параллельные тексты. — Школа Ильи Шальнова. http://shalnov-school.ru/parallels.html.
13. Sketch Engine: Корпус-менеджер. http://the.sketchengine.co.uk.
14. *Schmid H.* Probabilistic part-of-speech tagging using decision trees // Conference (International) on New Methods in Language Processing Proceedings. — Manchester, U.K.: UMIST, 1994. P. 44–49.
15. *Sharoff S., Nivre J.* The proper place of men and machines in language technology: Processing Russian without any linguistic knowledge // Компьютерная лингвистика и интеллектуальные технологии: По мат-лам Междунар. конф. «Диалог». — М.: РГГУ, 2011. Вып. 10(17). С. 591–604.
16. *Шарнин М. М., Сомин Н. В., Кузнецов И. П., Морозова Ю. И., Галина И. В., Козеренко Е. Б.* Статистические механизмы формирования ассоциативных портретов предметных областей на основе естественно-языковых текстов больших объемов для систем извлечения знаний // Информатика и её применения, 2013. Т. 7. Вып. 2. С. 92–99.

Поступила в редакцию 27.03.14

METHOD FOR EXTRACTING SINGLE-WORD TRANSLATION CORRESPONDENCES FROM PARALLEL TEXTS USING DISTRIBUTIONAL SEMANTICS MODELS

Yu. I. Morozova, E. B. Kozerenko, and M. M. Sharnin

Institute of Informatics Problems, Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation

Abstract: The paper deals with problems of corpus research of linguistic units. The task of extracting translation correspondences from a parallel corpus is defined. An overview of existing approaches to this task is provided. The paper focuses on the approach to extracting translation correspondences based on distributional semantics models. The paper describes the theoretical model developed by the authors as well as its software implementation. A test parallel corpus of patent texts in French and English was compiled for the purpose of this research. The paper provides results of an experiment aimed at extracting single-word translation correspondences from the test parallel corpus.

Keywords: extracting translation correspondences; alignment; parallel texts; parallel corpus; distributional semantics; vector space model

DOI: 10.14357/08696527140209

References

1. Brown, P. F., S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Comput. Linguistics* 19(2):263–311.
2. Och, F. J., and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Comput. Linguistics* 29(1):19–51.
3. Vogel, S., H. Ney, and Ch. Tillmann. 1996. HMM-based word alignment in statistical translation. *16th Conference on Computational Linguistics Proceedings*. Stroudsburg, PA, USA: Association for Computational Linguistics. 2:836–841.
4. Kozerenko, E. B. 2010. Lingvisticheskie fil'try v statisticheskikh modelyakh mashinnogo perevoda [Linguistic filters in statistical machine translation systems]. *Informatika i ee Primeneniya — Inform. Appl.* 4(2):83–92.
5. Masahiko, H., and T. Yamazaki. 1996. High-performance bilingual text alignment using statistical and dictionary information. *34th Annual Meeting of the Association for Computational Linguistics Proceedings*. Stroudsburg, PA, USA: Association for Computational Linguistics. 131–138.
6. Cendejas, E., G. Barcelo, A. Gelbukh, and G. Sidorov. 2009. Incorporating linguistic information to statistical word-level alignment. *14th Iberoamerican Conference on Pattern Recognition Proceedings*. Berlin, Germany: Springer. 387–394.
7. Lingvisticheskiy enciklopedicheskiy slovar' [Linguistic encyclopedia]. 1990. Ed. V. N. Jarceva. Moscow: Soviet encyclopedia.

8. Sahlgren, M., and J. Karlgren. 2005. Automatic bilingual lexicon acquisition using random indexing of parallel corpora. *J. Natural Language Eng.* (Special Issue on Parallel Texts). 11(3):327–341.
9. European patent agency. Available at: http://www.epo.org/index.html (accessed March 27, 2014).
10. Somin, N. V., I. P. Kuznetsov, V. G. Nikolaev, N. S. Solov'eva, and A. G. Mackevich. 2011. Metody ustraneniya neopredelennostey bloka leksiko-morfologicheskogo analiza pri izvlechenii znaniy iz tekstov estestvennogo jazyka [Methods of resolving ambiguity of lexical and morphological analysis in systems of knowledge extraction from natural language texts]. *Sistemy i Sredstva Informatiki — Systems and Means of Informatics* 21(2):97–115.
11. The Bilingual Sentence Aligner software tool. Available at: http://research.microsoft.com/en-us/downloads/aafd5dcf-4dcc-49b2-8a22-f7055113e656/ (accessed March 27, 2014).
12. The Make Bilingua software tool. Available at: http://shalnov-school.ru/parallels.html (accessed March 27, 2014).
13. The Sketch Engine corpus manager. Available at: http://the.sketchengine.co.uk (accessed March 27, 2014).
14. Schmid, H. 1994. Probabilistic part-of-speech tagging using decision trees. *Conference (International) on New Methods in Language Processing Proceedings*.
15. Sharoff, S., and J. Nivre. 2011. The proper place of men and machines in language technology: Processing Russian without any linguistic knowledge. *Komp'yuternaya lingvistika i intellektual'nye tekhnologii: Po materialam Mezhdunar. Konf. "Dialog"* [*Computational Linguistics and Intelligent Technology: Conference (International) "Dialogue"*]. Moscow: Publishing House of Russian State University for the Humanities. 10(17):591–604.
16. Sharnin, M. M., N. V. Somin, I. P. Kuznecov, Yu. I. Morozova, I. V. Galina, and E. B. Kozerenko. 2013. Statisticheskie mekhanizmy formirovaniya assotsiativnykh portretov predmetnykh oblastey na osnove estestvenno-jazykovykh tekstov bol'shikh ob"emov dlya sistem izvlecheniya znaniy [Statistical mechanisms of subject domains associative portraits formation on the basis of big natural language texts for the systems of knowledge extraction]. *Informatika i ee primeneniya — Inform. Appl.* 7(2):92–99.

Received March 27, 2014

Contributors

Morozova Yuliya I. (b. 1984) — scientist, Institute of Informatics Problems, Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; yulia-ipi@yandex.ru

Kozerenko Elena B. (b. 1959) — Candidate of Science (PhD) in linguistics, Head of Laboratory, Institute of Informatics Problems, Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; kozerenko@mail.ru

Sharnin Mikhail M. (b. 1959) — Candidate of Science (PhD) in technology, senior scientist, Institute of Informatics Problems, Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; keywen1@mail.ru

О ПРОБЛЕМАХ РЕАЛИЗАЦИИ СЕМАНТИЧЕСКОЙ ГЕОИНТЕРОПЕРАБЕЛЬНОСТИ В SEMANTIC WEB*

С. К. Дулин[1], *Н. Г. Дулина*[2], *Д. А. Никишин*[3]

Аннотация: Проблема семантической геоинтероперабельности заключается в обеспечении согласованного взаимодействия специалистов для решения задач, требующих совместного использования георесурсов, при условии адекватного понимания ими семантики этих георесурсов. Обеспечение семантической геоинтероперабельности предполагает, что разработка средств согласованного понимания геоданных должна осуществляться на основе сравнительного анализа существующих метасхем баз геоданных с учетом многофакторности взаимодействия пользователей и семантики, заложенной в пространственные онтологии и/или геотезаурусы и классификаторы. Ключевой задачей семантической геоинтероперабельности является создание единой концептуальной модели представления и согласованного понимания пользователями геоданных на основе интеграции пространственно-распределенной информации. Работа посвящена обсуждению понятия семантической интероперабельности, аспектам реализации семантической геоинтероперабельности географической информации (ГИ) и стандартам семантической геоинтероперабельности.

Ключевые слова: геоданные; семантическая геоинтероперабельность; онтологии; Semantic Web

DOI: 10.14357/08696527140210

1 Введение

В работе обсуждается текущее состояние интероперабельности ГИ и как оно может быть улучшено. Принятие стандартов — важное средство для формирования и развития взаимодействующих геопространственных данных и служб. Область ГИ должна основываться на немногих согласованных и хорошо продуманных стандартах и действиях, причем это цель не академического исследования, ведь именно рынок информационных услуг заинтересован в интероперабельности, что подтверждается недавними совместными действиями ведущих организаций по стандартизации — Консорциума геопространственной информации (Open Geospatial Consortium — OGC) и Технического комитета Между-

*Работа выполнена при поддержке РФФИ (проекты 14-07-00040 и 14-07-00785).

[1]Научно-исследовательский и проектно-конструкторский институт информатизации, автоматизации и связи на железнодорожном транспорте (ОАО НИИАС); Институт проблем информатики Российской академии наук, s.dulin@ccas.ru

[2]Вычислительный центр им. А. А. Дородницына Российской академии наук, ngdulina@mail.ru

[3]Институт проблем информатики Российской академии наук, dmnikishin@mail.ru

народной организации по стандартизации 211 (ISO's Technical Committee 211 (Geomatics, Geographic Information)).

Преграды для реализации интероперабельности серьезно ограничивают деятельность многих фирм и пользователей в области ГИ из-за увеличивающейся потребности в интеграции большого количества компонентов в информационных системах предприятий. Стандартизация протоколов, форматов и особенно интерфейсов может помочь преодолеть эти преграды, но это не все, что нужно сделать. Интероперабельность будет достигнута только в результате скоординированного усилия разнообразных участников: провайдеров данных, технологов и поставщиков услуг, организаторов стандартов и группы пользователей рынка ГИ.

Возможность совместного использования геопространственных данных была одним из основных требований начиная с разработки первой геоинформационной системы (ГИС). Существующие геоданные были получены независимо друг от друга разными организациями с помощью различных систем и используются во множестве приложений также независимо друг от друга. Характерны в этом смысле топографические данные, которые лежат в основе пространственного анализа, используемого для экологических исследований, критического положения оползней, размывов, управления транспортными потоками, безопасности, сельского хозяйства и т. д.

Несмотря на то что прошлые два десятилетия были очень продуктивными с точки зрения развития геоинтероперабельности, облегчающей совместное использование географических данных, геоинтероперабельность пока еще не стала общезначимой. Хотя стандарты, разработанные ISO/TC 211 и OGC, обеспечили основу для ее развития и в международных организациях (Canadian Geospatial Data Infrastructure (CGDI), National Spatial Data Infrastructure (NSDI), Infrastructure for Spatial Information in Europe (INSPIRE), Global Spatial Data Infrastructure (GSDI)) возникли пространственные инфраструктуры данных, реализация геоинтероперабельности все же находится на начальной стадии.

С другой стороны, за эти два десятилетия произошло внушительное развитие Интернета и Web. Организация Web началась с очень простой публикации соединенных между собой веб-страниц. Теперь Web составлена из порталов, сервисов, данных, документов, видео, музыки, перманентно пополняясь коллекциями данных различных видов типа wikis, карты Google, OpenStreetMap и т. д. В настоящее время Web совершенствуется до Semantic Web (или Web 3.0), эволюционируя от веб-документов до веб-данных, превращаясь в международную открытую базу данных. Все это внушает определенный оптимизм в отношении реализации геоинтероперабельности.

Статья состоит из восьми разделов. Раздел 2 посвящен понятию интероперабельности. В разд. 3 представлены уровни интероперабельности. В разд. 4 обсуждается понятие онтологии, являющееся основополагающим понятием в исследовании семантической геоинтероперабельности. Раздел 5 вводит основные понятия, связанные с семантической геоинтероперабельностью. Далее в

разд. 6 рассматриваются стандарты семантической интероперабельности в контексте ГИ, которые поддерживают геоинтероперабельность для географических информационных и пространственных инфраструктур данных. Раздел 7 посвящен Semantic Web, которая обеспечивает важную поддержку семантической геоинтероперабельности применительно к пространственным инфраструктурам данных и представляет собой симбиоз семантической геоинтероперабельности и семантической сети. Раздел 8 содержит заключительные замечания.

2 Понятие интероперабельности

Существительное «интероперабельность» означает способность одной системы использовать части другой системы (Словарь Merriam-Webster, http://www.m-w.com) без специального усилия со стороны клиента; т. е. она превышает возможности коммуникации (http://www.dictionary.com). Здесь с позиции информатики основная суть определения — это способность использовать информацию, полученную в результате обмена. Международная организация по стандартизации ISO 19119 предлагает следующее определение (http://www.wmo.int/pages/prog/www/TEM/ET-WISC-I/ISO_191xx.doc):

«Интероперабельность представляет собой способность соединяться, выполнять программы или передавать данные среди различных функциональных модулей способом, который не требует, чтобы пользователь имел знания о характеристиках этих модулей».

Два компонента A и B (рис. 1) могут взаимодействовать (являются интероперабельными), если A может послать основанные на взаимном понимании A и B запросы Q для использования сервисов, находящихся у B, и если B может подобным образом возвратить взаимно понятные ответы R к A.

Это означает, что две интероперабельные системы могут взаимодействовать совместно для выполнения задачи.

Для географической области применимо следующее описание термина «географическая интероперабельность» (http://www.wmo.int/pages/prog/www/TEM/ET-WISC-I/ISO_191xx.doc):

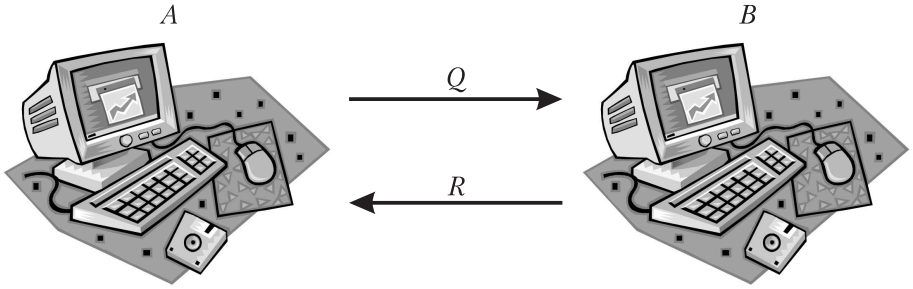


Рис. 1 Интероперабельность

«Географическая интероперабельность — это способность информационных систем к (1) свободному обмену всеми видами пространственной информации о Земле и об объектах и явлениях на, выше и ниже поверхности Земли и (2) совместному сетевому использованию программного обеспечения, предназначенного для управления такой информацией».

Следует отметить, что это определение не предполагает, что каждый компонент использует один и тот же формат данных (совпадение форматов соответствует обычно неправильному восприятию многими людьми интероперабельности), а скорее провозглашает способность понимать формат(ы) друг друга.

На техническом уровне интероперабельность включает коммуникацию, определенную соответствующим протоколом коммуникации, аппаратными средствами, программным обеспечением и уровнями совместимости данных. В то время как вышеупомянутое можно было бы назвать обеспечением *синтаксической* интероперабельности в смысле передачи параметров, *семантическая* интероперабельность имеет дело со знаниями проблемной области, необходимыми для сервисов информатики, чтобы «понять» мотивации и креативы друг друга. При этом для разрозненных сервисов необходимо взаимодействовать, создавая сервисные цепочки обслуживания в сети. Важность понятия интероперабельности постоянно возрастает для продуктов информационной технологии в рамках концепции «сеть — это компьютер», которая становится действительностью.

Интероперабельность имеет много значений, включая понятия коммуникации, обмена, кооперации и совместного использования ресурсов различными системами. Таким образом, сущность интероперабельности — в реализации отношений между системами, где каждое отношение — способ коммуникации, обмена, кооперации и совместного использования ресурсов. Современная ГИС может быть составлена из нескольких компонентов аппаратных средств, программного обеспечения, ресурсов, правил управления, процедур и людей, которые хранят, обрабатывают и обеспечивают доступ к ГИ. Эти компоненты традиционно предоставляют пользователям гомогенный подход, основанный на обработке отдельного продукта ГИС и использовании технологии системы как фактического стандарта. В противоположность этому ГИС, обладающая интероперабельностью, обрабатывает среды, в которых разнообразие тех же самых компонентов сосуществует и взаимодействует.

В контексте ГИ интероперабельность непосредственно связана с ГИС. Геоинформационная система представляет собой основную среду для реализации геоинтероперабельности, потому что в ней в центре внимания пространственные данные, которые более сложны, чем обычные текстовые данные, сохраненные в реляционных базах данных. Характеристики ГИ не предусмотрены для отображения реляционной моделью, а объектная модель является слишком общей для независимого выполнения операций любого вида, что препятствует интероперабельности. По общему мнению, многие из проблем геоданных в области ГИ были созданы самими специалистами этой области, так как каждый разработчик стремился создать новые способы представления геометрических объектов,

изображений и атрибутивных данных. Эта ситуация сильно отличается от гомогенных способов представления других данных, например данных в электронных таблицах и в базах данных. Проблемы интероперабельности ГИ выглядят еще более серьезными, если добавить к присущей ГИ сложности геоданных институциональные и юридические осложнения [1]. Этим, по-видимому, и объясняется существование довольно большого числа необходимых стандартов.

В основе реализации интероперабельности любого типа лежит коммуникация. Исследование коммуникации началось с Шеннона и Винера — основоположников теории информации и кибернетики. Кибернетика, по существу, изучает процессы коммуникации и управления, которые используются в системах для поддержания порядка, организации и равновесия. Информация соответствует содержанию, которым система обменивается с другими системами. Хорошее выполнение задания напрямую зависит от полученной информации и коммуникации с другими системами. Информационный обмен между системами — это вопрос эффективной коммуникации и управления. Принято считать, что процесс коммуникации состоит из трех компонентов: источника, сообщения и адресата.

Если адресат не имеет никакого общего знания с источником, коммуникация не может быть эффективной, и поэтому адресат не может декодировать и интерпретировать сообщение должным образом. Например, два человека должны иметь общее знание о языке, который они используют, чтобы взаимодействовать. В противном случае коммуникация невозможна. Помимо их знания о языке они должны также совместно использовать определенное число подобных понятий из некоторого терминологического словаря.

3 Уровни интероперабельности

Исследования последних лет в области интероперабельности указывают на необходимость создания моделей интероперабельности, которые могут гарантировать, что интероперабельность устанавливается между системами в соответствии с различными целями и контекстами [2].

Существует несколько подходов к формированию ГИС посредством моделей интероперабельности [3]. Каждый подход имеет преимущества и недостатки относительно достижения интероперабельности в определенном контексте. Основные преимущества моделей интероперабельности — это возможность (а) определения общего словаря, который обеспечивает единообразие семантики и возможность анализа; (б) альтернативы предложений относительно структуры решений и, наконец, (в) оценки новых идей и добавления различных опций. Без сомнения, самой неисследованной проблемой ГИС, обладающей интероперабельностью, все еще остается ликвидация разрыва между различными моделями, что обеспечило бы создание объединенного метода, учитывающего сильные стороны и слабости каждой отдельной модели при их интеграции. В настоящее время каждая модель интероперабельности в ГИС определяет общую таксономию, которая поддерживает различные цели использования, достигая интероперабельности в разных

Рис. 2 Уровни интероперабельности

контекстах. Слои, измерения, уровни или области — это понятия, которые обычно определяют такую таксономию. Они зависят от типа контекста и применяемой модели интероперабельности для формирования ГИС с интероперабельностью.

Примерами моделей интероперабельности, которые были успешно применены вне специфики ГИС-области, являются C4ISR Architectures Working Group's Levels of IT Systems Interoperability (LISI) model, the Enterprise Interoperability Maturity Model (EIMM), the Organisational Interoperability Maturity Model (OIMM) и the Organisational Interoperability Agility Model (OIAM). Две модели — the Levels of Conceptual Interoperability Model (LCIM) и the Intermodel5 использовались в ГИС-области. Все эти модели в основном используются на самых высоких уровнях организационной интероперабельности из традиционных семи уровней: уровня нулевой интероперабельности, технической, синтаксической, семантической, прагматической, динамической и концептуальной интероперабельности (рис. 2) [3].

На техническом уровне интероперабельности обычно рассматриваются представление и обмен данными, доступность и характеристики безопасности типа протоколов, интерфейсов, форматов документа, кодирования данных, а также меры доступности и решения безопасности. Здесь важны технические аспекты интероперабельности распределенной вычислительной среды, сетей коммуникации, непосредственно технологий и распределенных платформ вычисления. Техническая интероперабельность в основном позиционируется как связь между компьютерными системами и сервисами.

Технический уровень интероперабельности — это основа, которая позволяет провайдерам ГИ поставлять данные и сервисы с незначительными затратами, получая прибыль от их использования.

Синтаксический уровень интероперабельности обеспечивает общую структуру обмена информацией, в которой применяется установленный формат данных. В контексте ГИС синтаксическая интероперабельность определяется как спецификация общих форматов сообщения для обмена пространственными данными, образцами и связями. На этом уровне интероперабельности потребность транслировать метаданные от одного прикладного контекста к другому — важное

требование, которому уделено недостаточно внимания в области ГИС. Исследование качества метаданных полагается на понимание, насколько метаданные соответствуют существующему обмену данными на синтаксическом уровне для диапазона возможных приложений во взаимодействующей ГИС. Такое исследование обычно начинается с идентификации атрибутов кандидата в метаданные и продолжается анализом взаимозаменяемости метаданных.

Семантическая интероперабельность связана со значением совместно обрабатываемой информации или, другими словами, со смыслом данных, воспринимаемым разными людьми. Различия в информационном контексте происходят обычно из-за разных значений одного и того же реального объекта, который сохранен в разных базах данных.

Очевидно, что всегда будет возникать проблема, связанная с тем, что географическое пространство может иметь больше чем одно описание в базах данных ГИС, которая обслуживает различные дисциплины, приводя, как следствие, к семантической разнородности. Другая проблема состоит в том, что разные пользователи ГИС должны быть в состоянии понять значение информации, полученной в результате обмена. В этом случае семантическая интероперабельность должна обеспечить адекватное понимание значения полученной информации любым другим приложением. Известно несколько попыток создания стандартизированных таксономий на основе Географического стандарта хранения ГИ для интеллектуальных транспортных систем [4].

К прагматическому уровню относятся намерения, обязанности и последствия сформулированных утверждений. Прагматическая интероперабельность может быть доведена до такой степени, что пользователи взаимодействующих сервисов будут иметь совместные намерения, обязанности и результаты в рамках взаимодействующих сервисов и информационного обмена. В контекстной прагматической интероперабельности особенно важен тот факт, что все стороны, вовлеченные в обеспечение и использование сервисов, несут определенную ответственность. Этот уровень может быть достигнут, когда взаимодействующие системы осведомлены об используемых методах и процедурах. Другими словами, использование данных или контекста их приложения должно быть понятно участвующим системам. Основная исследовательская проблема здесь связана с тем, как обеспечить механизмы прагматического уровня интероперабельности в сервисно-ориентированной архитектуре (SOA). Сервисно-ориентированная архитектура представляет функции модулями или сервисами, которые разработчики делают доступными в сети, чтобы пользователи могли комбинировать их и многократно использовать в ГИС-приложениях [5]. Разработка сетевых и базируемых на SOA сред на прагматическом уровне в последнее время стала критически важной.

Динамический тип интероперабельности имеет место, когда системы в состоянии воспринимать изменения состояния, которые происходят с предположениями и ограничениями, сделанными в течение долгого времени, и в состоянии использовать эти изменения в своих интересах.

Динамическая интероперабельность может быть рассмотрена в этом случае в двух аспектах: динамическая интероперабельность данных и динамическая сетевая интероперабельность. Динамическая интероперабельность данных должна гарантировать обмен ими и интеграцию считываемых данных с другими видами информационных ресурсов. С другой стороны, цель сетевой интероперабельности — интеграция сетевых компонентов, которые должны обмениваться данными и взаимодействовать с информацией, поставляемой другими компонентами или внешними сетями. Компоненты и сети должны совместно использовать память, коммуникацию и ресурсы считывания. Поэтому динамическая интероперабельность необходима, чтобы выполнить коммуникацию между сетевым шлюзом и пользователями, а также в сетях, чтобы обмениваться сообщениями и обрабатывать сетевые сообщения.

Установлено, что концептуальный тип интероперабельности может быть достигнут, когда концептуальная модель зарегистрирована техническими методами так, чтобы она могла интерпретироваться и оцениваться третьим лицом. Если концептуальные модели (т. е. предположения и ограничения значимой абстракции действительности) зафиксированы, то достигнут самый высокий уровень интероперабельности.

4 Онтологии

Сегодня ведущей парадигмой структурирования информационного контента являются онтологии, или иерархические концептуальные структуры, которые формируются аналитиком на основе изучения и структурирования потоков информации, документов, протоколов извлеченных знаний и других источников. С методической точки зрения это один из наиболее систематических и наглядных способов структурирования и формализации знаний.

Онтология в информатике согласно современным толкованиям является «эксплицитной спецификацией концептуализации предметной области» [6], но с определенными ограничениями в зависимости от области интересов и должна включать словарь терминов и некоторые спецификации их значений. Использование онтологий способствует созданию адекватных концептуальных моделей, обеспечивая качественное, контролируемое информационное интегрирование.

Онтологии — содержательные теории, которые включают общий набор распространяемых фактов, чье основное назначение — идентифицировать определенные классы объектов и отношений, которые существуют в некоторой части предметной области. Таким образом, неформально определенные онтологии — это соглашения об общедоступной концептуализации. Формальное определение основывалось бы на том, что онтология является (возможно, неполной) аксиоматизацией допустимых прикладных моделей. Другими словами, онтология состоит из основного словаря и отношений, используемых для описания некоторых аспектов действительности, включая ряд аксиом, связанных с предполагаемым значением словаря.

Несмотря на большой накопленный опыт сбора, обработки и анализа данных и широкое применение изображений поверхности Земли, онтологический статус таких изображений остается открытой проблемой. Удивительно, но не всегда просто ответить на элементарный вопрос: «Что находится на изображении земной поверхности?» или перефразировать тот же самый вопрос по-другому: «Что является онтологическим статусом информационного содержания изображений, полученных в результате дистанционного сканирования или фотосъемки поверхности Земли?»

Для ответа на этот вопрос необходимо решить как минимум две проблемы: (1) разработать концептуальное основание для всех типов компьютерных представлений географического пространства, включая изображения, векторные данные, информацию о расположении и цифровые модели ландшафта; (2) разработать технологию ГИС, которая могла бы объединить изображения без разрывов в пространственную базу данных на основе понимания онтологического статуса изображений, полученных в результате дистанционного сканирования. Такая интеграция особенно актуальна в контексте нового поколения пространственных информационных систем, которые, как ожидается, обеспечат представление и использование онтологий.

Наиболее широко принятая концептуальная модель данных для пространственной информации предполагает, что географическая действительность представлена или как полностью определимые сущности (объекты), или как непрерывное пространственное изменение (область, поле). Модель объекта представляет мир как поверхность, занятую дискретными объектами с геометрическим представлением и описательными признаками. Модель на основе парадигмы поля представляет географическую действительность как ряд пространственных распределений в географическом пространстве. Хотя эта простая дихотомия была неоднократно подвергнута критике, она оказалась полезной системой взглядов и была принята с некоторыми уточнениями при проектировании технологий ГИС [7].

Дихотомия объекта с непрерывными характеристиками — характерное географическое понятие, не предназначенное, тем не менее, обеспечивать поддержку специфики семантики различных типов пространственных данных. Этот недостаток заставил многих исследователей перенести акцент на использование онтологий как средства диссеминации знаний в группах пользователей с различными интересами, тем самым улучшая функциональную совместимость с различными базами геоданных. При этом диссеминация знаний напрямую зависит от успешности трансформации неявных знаний в явные. Формализация неявных знаний представляет большую сложность, поскольку они во многих случаях носят личностный характер, контекстно зависимы и трудно выявляемы.

На рис. 3 приведена классическая схема преобразования неявных знаний в явные, состоящая из четырех основных этапов: обобществления, формализации, диссеминации и усвоения [8]. Под обобществлением понимается обобщенный процесс выработки неявных знаний. Формализация предполагает изложение

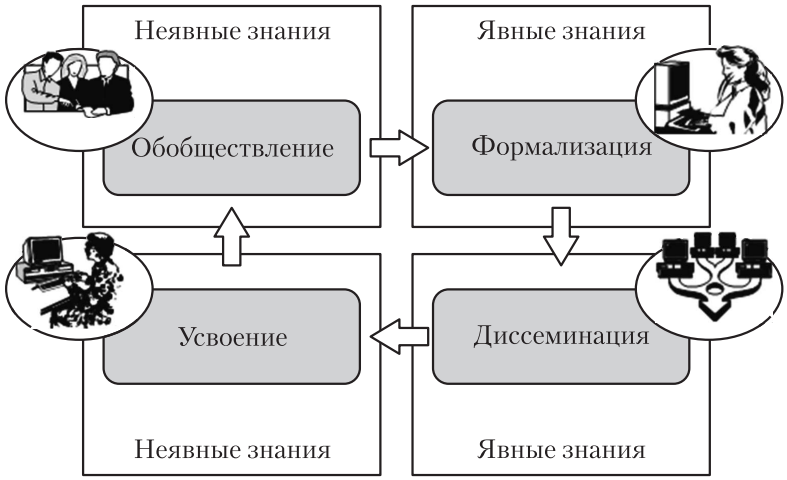


Рис. 3 Преобразование знаний

своих знаний в той или иной явной, наглядной или читаемой форме. Чтобы ознакомить с результатами своего труда других специалистов, необходимо преобразовать знания из одной формы представления в другую и распространить их по доступным каналам связи. Этот процесс назван диссеминацией. На четвертом этапе происходит изучение полученного знания, его осознание и усвоение.

Следует заметить, что специфика географического мира в достаточной мере определяет параметры создания онтологий. Чтобы адекватно представлять географический мир, необходимо иметь компьютерные представления географических знаний (в первую очередь — изображений), которые способны фиксировать не только описательные атрибуты пользовательских концепций, но также и описывать геометрические и позиционные компоненты этих концепций. Эти представления также должны фиксировать пространственные и временны́е зависимости между экземплярами этих концепций. В отличие от случая обычных информационных систем, большинство пространственных и временны́х зависимостей не представлены в ГИС и чаще всего могут просто выводиться путем использования различных географических функций. Поэтому обязательно должна быть привнесена дополнительная семантика в схему географического приложения, семантические спецификации которой, являющиеся частью онтологии этого приложения, зафиксированы разработчиком модели. Новое поколение информационных систем должно обладать способностью обрабатывать семантическую неоднородность, возникающую в результате использования разнородных источников информации.

Использование множественных онтологий становится основной особенностью современных информационных систем, если в них предполагается поддержка семантики при интеграции информации. Онтологии могут фиксировать семантику

информации, могут быть представлены в формальном языке и могут также использоваться, чтобы хранить связанные метаданные, допуская, таким образом, семантический подход к информационному интегрированию [9].

Представление пользовательских онтологий прикладной области рассматривается как существенная часть фиксации концепций информационного пространства [10]. Исследование онтологического статуса пространственных типов данных — наиболее актуальное направление в геоинформатике. В [11] впервые было введено понятие ГИС, управляемой онтологиями, призванной обеспечивать пользователей ГИ возможностью достигнуть соглашения по основным сущностям географического мира. Идея управления с помощью онтологий заключается в том, что существенная часть географического знания зафиксирована процедурами, которые извлекают информацию из пространственных наборов данных, т. е. для этого необходимо создать онтологии не только для объектов некоторой области, но также и для намеченных действий, которые выражены процедурами, применимыми к набору данных, предназначенному для извлечения знания.

5 Семантическая геоинтероперабельность

Интероперабельность является показателем эффективной коммуникации между системами. Реализация коммуникации связана как с синтаксическими, так и с семантическими проблемами. Однако географические базы данных и представления пользователями реальных явлений характеризуются большой гетерогенностью, которая препятствует эффективной геоинтероперабельности. Как правило, гетерогенность декомпозируется на четыре уровня: системный, синтаксический, структурный и семантический.

Базы данных часто располагаются в разных системах, и для поддержки геоинтероперабельности требуется установление взаимосвязи между ними. С развитием сетей и протоколов коммуникации, таких как Ethernet, Transmission Control Protocol/Internet Protocol (TCP/IP), Remote Procedure Call (RPC), File Transfer Protocol (FTP), Hypertext Transfer Protocol (HTTP) и др., стала возможна взаимосвязь систем, работающих под различными операционными системами (Linux, Windows, Mac OS и др.). Кроме того, базы данных различных типов также могут быть связаны с целью совместного использования данных через приложения посредством интерфейсов Open Database Connectivity (ODBC), Java Database Connectivity (JDBC) и др.

Синтаксическая гетерогенность зависит от способа, которым данные представлены физически, используемых знаков и их порядка в сообщении. Синтаксис устанавливает знаки и правила, которые определяют порядок знаков в сообщении. Форма сообщения имеет здесь основное значение по сравнению с его содержанием, на основе анализа которого декодируются знаки сообщения. Форма должна соответствовать формату обмена данными, чтобы обеспечить доставку данных, например, как в Geography Markup Language (http://www.opengeospatial.org/standards/gml). Синтаксическая гетерогенность учи-

тывает также и различные формы представления данных: например, ГИ может принять форму покрытия данных с координатной привязкой и векторных данных.

Структурная гетерогенность определяется различиями при моделировании данных. В некоторых базах данных что-то может быть описано как понятие, а в других — как характеристика понятия. Например, «Измайловский парк» может быть определен как понятие в одной базе данных, тогда как это понятие может быть описано как характеристика «Восточной лесопарковой зоны г. Москвы» в другой. Структурные конфликты в ГИ могут возникать на следующих уровнях: понятие, свойство, геометрия и временной характер.

Семантическая гетерогенность связана с различиями значений понятий. Понятие является результатом связей со знаком и объектом. Различия в когнитивных моделях разных людей, которые связывают идентичные знаки с различными явлениями и одни и те же явления с различными знаками, иллюстрируют проблему семантической гетерогенности. Контекст играет здесь существенную роль, так как когнитивные модели разрабатываются в определенных контекстах. Фактически именно контекст задает семантику реальных явлений; он определяет, как явления воспринимаются и абстрагируются, и затем влияет на их определение в терминах понятий. Необходимо учитывать контекст и при рассуждении, анализируя семантическую гетерогенность и подобие понятий.

Геоинтероперабельность можно представить как двунаправленный механизм в противоположность конвейеру, обрабатывающему информацию только в одном направлении (от источника к адресату). Установление семантической геоинтероперабельности выходит за пределы простой возможности получить доступ к информации географических баз данных на дисплее или напечатанной на бумаге. Оно требует больше времени, заранее должен быть известен точный словарь географических баз данных, чтобы получить соответствующую информацию. Но самое существенное, что установление семантической геоинтероперабельности предполагает, что пользователи и провайдеры должны иметь релевантное понимание семантики запросов и ответов. В контексте Semantic Web такая возможность становится все более и более доступной.

Моделирование семантики глубоко внедрено в структуру геоинтероперабельности и, таким образом, обеспечивает исчерпывающее описание семантической геоинтероперабельности в целом, которая лежит в основе развития семантической пространственной инфраструктуры данных и Semantic Web геопространственной информации.

Геоинтероперабельность представляет собой основу для развития и реализации пространственных инфраструктур данных (Spatial Data Infrastructures — SDIs) [12]. Цель SDIs состоит в том, чтобы координировать полезный обмен и совместное использование ГИ с использованием соответствующих сервисов. SDIs — это средства разработки функциональной совместимости для ГИ. SDIs составлены из пяти элементов: методики, технологии, стандарты, человеческие ресурсы и релевантные действия, требуемые для сбора, обработки, управления, доступа, поставки и использования ГИ. SDIs были разработаны на основе ре-

ференсной модели для Open distributed processing (RM-ODP) [13]. RM-ODP структурно объединяет пять разделов: раздел предприятия, информационный раздел, вычислительный раздел, технический раздел и раздел технологии. В контексте SDIs раздел предприятия описывает цели и возможности, методики, обязанности и бизнес-процессы SDIs. Информационный раздел, по существу, посвящен информации, доступной через SDIs и необходимой для семантической геоинтероперабельности. Вычислительный раздел касается функциональной декомпозиции SDIs в сервисы с интерфейсами и операциями. Этот раздел представляет большой интерес для определения семантических компонентов и сервисов. Технический раздел главным образом связан с взаимодействием между данными, сервисами и системными взаимосвязями. Наконец, раздел технологии относится к определенно выбранной технологии для реализации SDIs.

Первые три из указанных разделов особенно важны с точки зрения семантической геоинтероперабельности. Именно в разделе предприятия цель достижения семантической геоинтероперабельности должна быть четко сформулирована. Раздел предприятия должен идентифицировать любой репозиторий, который должен участвовать в SDIs, а также поставщиков данных, которые помогут пользователям и провайдерам в поиске соответствующих данных, релевантных их словарю и семантике. Вряд ли все пользователи могут знать заранее точный словарь и семантику, используемую геоинформационными источниками, доступными в SDIs. Поэтому пользователи должны быть в состоянии взаимодействовать с SDIs, используя свой собственный словарь, и находить данные, которые соответствуют их определенной цели. Следовательно, информационный раздел должен включать необходимые информационные компоненты, чтобы обеспечить семантические запросы (т. е. запросы, сделанные в словаре пользователя и правильно интерпретируемые информационным сервером). Онтологии поэтому должны быть объединены как часть информационного раздела, обеспечивая фундаментальное знание для рассуждения, интерпретации запроса и выдачи соответствующих ответов. Вычислительный раздел нуждается также в определенном внимании, чтобы можно было решать семантические проблемы определения семантических интерфейсов, связанных с онтологиями, включая операции и функции логического вывода, которые помогают интерпретации запросов и ответов. Этот раздел включает также кодирование онтологий, чтобы устанавливать с помощью интерфейса связь с семантическими сервисами.

Учет семантики на раннем этапе разработки SDIs облегчает их дизайн и позволяет идентифицировать онтологии и требуемые семантические сервисы.

6 Геоинформационные стандарты для семантической геоинтероперабельности

Стандарты — это установленное согласие, представленное как модель или пример (http://www.m-w.com/). В контексте информационных систем они

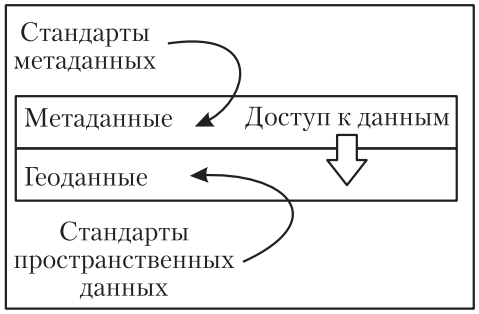


Рис. 4 Интероперабельность и стандарты

подразумевают соглашения о представлении информации в таком виде, чтобы компьютеры могли «понять» общий язык, обеспечивающий понимание и сотрудничество. Стандартизация касается главным образом технического и семантического уровней интероперабельности. Как видно из рис. 4 (http://www.ec-gis.org/etemii), помимо стандартов, имеющих дело с манипуляцией географическими данными (пространственные *стандарты*, обеспечивающие интерфейсы для стандартных методов), существуют также и стандарты для доступа к географическим данным из метаданных и стандарты метаданных, которые используются для каталогов, описывающих содержание информационных ресурсов пространственных данных. Эти ресурсы необходимы, чтобы можно было найти геоданные в сети, и поэтому они должны быть в стандартных форматах (*стандартах метаданных*).

Стандарты используются также, чтобы корреспондироваться с другими ресурсами типа открытых веб-сервисов (www.opengis.org/ows) для совместного использования или формирования цепочки сервисов.

Стандартизация в области цифровой ГИ необходима для обеспечения гарантии, что пространственная информация может свободно распространяться в глобальной среде. Разработка международных стандартов продвигается чрезвычайно быстрыми темпами. Здесь, как уже отмечалось выше, определились два главных разработчика: OGC и ISO/TC211 с широко известными результатами совместного сотрудничества.

TC 211 — один из множества технических комитетов, зарегистрированных в ISO (Международной организации по стандартизации). Он был сформирован для развития стандартизации в области цифровой ГИ. Основная цель TC 211 — установить структурированный набор стандартов для информации относительно объектов или явлений, которые прямо или косвенно связаны с местоположением относительно Земли. Эти стандарты могут специфицировать ГИ, методы, инструментальные средства и сервисы для организации данных (включая определение и описание), приобретения, обработки, анализа, доступа, представления и передачи таких данных в цифровой/электронной форме между различными пользователями, системами и местоположениями. TC 211 связан с разработкой набора стандартов, допускающих геоинтероперабельность. Эти стандарты поддерживают понимание и использование ГИ, улучшают ее доступность, интеграцию, совместное использование и разработку SDIs. Все это обеспечивает фундаментальную структуру и семантику для описания и представления ГИ. В их основе лежит референсная мо-

дель, основанная на RM-ODP, в которой большое внимание уделено семантике.

Open Geospatial Consortium, Inc. — некоммерческая организация, основанная в 1994 г. для поддержки разработок интероперабельности систем, которые обрабатывают геоданные, а также между такими системами и функционирующими вычислительными системами (http://www.Opengis.org/info/brochure/brochure0599.pdf). Деятельность OGC по развитию технологий существенно повлияла на глобальную организацию геоданных и создание стандартов обработки геоданных и продемонстрировала преимущества открытых технологий ГИС, которые объединяют обработку геоданных с распределенной архитектурой и интернет-вычисления.

Деятельность OGC фокусируется на выполнении двух программ: программы спецификации и программы интероперабельности, с явным приоритетом в отношении последней.

Программа спецификации — программа установления согласия среди членов консорциума, обладающих правом голоса. Во многом эта программа подобна процессу создания международных стандартов. Основные различия здесь связаны с организацией самого процесса согласования, подхода к решению, кто будет воздействовать на разработку этого согласия со стороны основной группы сотрудников OGC, разработчиков и независимых консультантов.

Программа интероперабельности развивается относительно недавно. В ней установлена необходимость гарантировать заинтересованному лицу, что его требования будут приняты во внимание на ранней стадии и что они будут влиять на конечные результаты, а согласование семантики концепции определено как основная веха всей работы: после согласования семантики концепции ее новая спецификация проходит через программу спецификации. В контексте этой формулировки может быть замечено, что взаимосвязанные требования интероперабельности множества сотрудничающих заинтересованных лиц должны быть удовлетворены в рамках конкретного проекта, в результате чего заинтересованные лица становятся интероперабельными [14].

Все понятия, определенные в ISO географических информационных стандартов, составляют внушительную онтологию для географической информационной области. Понятия формализованы графически с помощью универсального языка моделирования (UML). Хотя UML имеет достаточную выразительность для определения понятий с ISO/TC 211, получающиеся диаграммы не предназначены, чтобы быть машиночитаемыми и обрабатываемыми, и не могут использоваться непосредственно, чтобы поддержать логический вывод на основе онтологии.

7 Semantic Web геопространственной информации

В 2001 г. была выдвинута идея Semantic Web [15]. В ее основе лежало предложение модернизировать Web от уровня документов до уровня данных и

информационного моделирования. В Semantic Web данные должны быть понятными и обрабатываемыми, и поэтому Semantic Web должна быть способной к ответу на вопросы, в отличие от простой передачи документов или веб-страниц, соответствующих определенным критериям ключевого слова. Semantic Web — это Web независимых от приложения данных, которые могут быть составлены из многих источников; данных, размещенных в системе классов; данных, которые являются частью информационной системы, объединенной посредством онтологий. Данные в Semantic Web намного более адаптивны по сравнению с Web. На самом низком уровне данные могут принять форму документов и записей в электронных таблицах или базах данных. Средний уровень может быть расширяемым языком разметки (XML, eXtensible Markup Language) документов с соответствующими словарями. На верхнем уровне данные хранятся вместе с онтологиями и с возможностями вывода нового знания. Онтологии весьма существенны для Semantic Web. Они определяют значения данных и описывают их в формате, который является вычислимым и читаемым приложениями. Semantic Web непосредственно связана с установлением геоинтероперабельности данных в Web.

Онтологии весьма существенны для Semantic Web. Они определяют значения данных и описывают их в формате, который является вычислимым и читаемым приложениями. Таким образом, когда приложения используют данные, они могут использовать врожденную семантику данных и в то же самое время сделать их более достоверными. Эта дополнительная возможность облегчает интеграцию гетерогенных данных, зафиксированных различными группами пользователей, на основании подобия семантики.

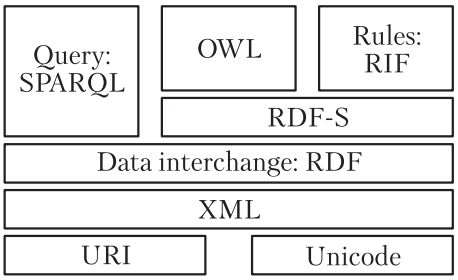


Рис. 5 Архитектура Semantic Web

Semantic Web требует логических утверждений, классификации понятий, формальных моделей, правил и протоколов безопасности и доверия. Архитектура Semantic Web была подробно описана во многих публикациях [16]. Как показано на рис. 5 [16], она основана на Uniform Resource Identifiers (URI), универсальном наборе символов (Unicode), и XML.

URIs — строки символов, которые служат уникальными идентификаторами, чтобы указать ресурс в сети (физический или абстрактный). URIs имеют определенный синтаксис в соответствии со стандартами 3986. URIs включают как Uniform Resource Locators (URLs), так и Uniform Resource Names (URNs). URL идентифицирует место, где ресурс доступен в Интернете и как добраться до него.

Unicode — промышленный стандарт, который используется как основа для представления текста в Semantic Web. Он допускает представление тексто-

вых строк и манипуляцию ими на разнообразных языках через стандартизацию символьного набора и символьных кодов, которые соответствуют стандарту ISO 10646 Информационных технологий — Universal Multiple-Octet Coded Character Set (UCS). Уникальное число определяет каждый символ независимо от системы, приложения и языка. Соответственно, Unicode позволяет системам и приложениям обмениваться символьно-ориентированной информацией и просматривать информацию веб-страницы без потери информации.

XML — синтаксическая основа Semantic Web. На этой основе консорциумом World Wide Web был разработан стек синтаксисов и словарь XML: RDF, RDF-S, Web Ontology Language (OWL), SPARQL Protocol and RDF Query Language (SPARQL) и Rules Interchange Format (RIF). Первым был разработан Resource Description Framework (RDF). RDF допускает представление информации об определенных ресурсах, которые существуют в сети. Он использует обычно тройную структуру, называемую «субъект–предикат–объект», похожую на семантический треугольник Фреге (концепт–знак–денотат).

Схема RDF (RDF-S) определяет классы и свойства, требуемые для описания классов, свойств и других ресурсов. Она основана на RDF. Ресурсы RDF-S служат для создания прикладных или определенных пользовательских словарей RDF и, кроме того, для создания классов специальных данных.

SPARQL, который включает протокол SPARQL и язык запросов RDF, является языком Semantic Web, обеспечивающим доступ к источникам данных RDF и онтологиям на языке OWL.

Язык OWL был разработан для представления знаний. Он позиционируется как развитие существующих языков DAML+OIL (DAML: DARPA Agent Markup Language; OIL: Ontology Inference Layer) и основан на RDF и RDF-S. OWL — фундаментальная часть Semantic Web для выполнения рассуждений, интерпретации и вывода. Онтологии поддерживают программных агентов для интерпретации поступающих элементов данных. Программные агенты могут автоматически приписывать значения поступающим сообщениям (запросам или данным). Такая функция реализуется сопоставлением онтологий или оценкой подобия.

В пространственной области подобие между геометрическими конструкциями (точка, линия, поверхность и тело) обычно идентифицируется матрицей пересечений внутренней области, границы и внешней области двух геометрических объектов. Хотя количественная оценка семантического подобия достаточно интересна, качественная оценка семантического подобия все же ближе к когнитивному рассуждению. Подход на основе геосемантической близости — это использование матриц пересечений для оценки семантического подобия между контекстами двух геопространственных понятий. Рассуждение и вывод могут привести к широкому диапазону возможностей, связанных с геоинтероперабельностью: обнаружение данных, ответ на запрос, композиция геоданных из многих источников, использование геоданных в разных областях, интеграция данных и т. д.

Формат обмена правилами (RIF) относится к ряду форматов World Wide Web и определяет различные виды систем правил для поддержки функциональной совместимости среди языков правил, а также для обмена правилами в системах, основанных на правилах в Semantic Web.

В составе Semantic Web не только данные, но также и веб-сервисы. В настоящее время использование и взаимодействие веб-сервисов все еще требует участия человека, хотя в перспективе Semantic Web могла бы облегчить взаимодействие с веб-сервисами, используя семантику. Semantic Web может поддерживать веб-сервисы, чтобы автоматизировать их открытие и оптимизировать их состав, обеспечивая их взаимодействие с минимальным человеческим участием.

Идея создания геопространственной Semantic Web впервые была представлена в 2002 г. [17]. Она должна расширить понятие Semantic Web, улучшив семантическую функциональную совместимость ГИ в Web.

За это время усилия по стандартизации ISO/TC 211 и OGC и развитие геоинформатики обеспечили в большой степени основу для создания геопространственной Semantic Web. Международные стандарты ISO/TC 211 определили онтологию геопространственных понятий, которые являются независимыми от приложений. Эта онтология — основа описания ГИ, которая включает понятия для описания геометрии, топологии, временно́й информации, пространственных систем справочной информации, особенностей, характеристик, поведений, отношений, качества, метаданных, сервисов (позиционирование, изображение, местоположение и т. д.); образы, решетки, охват, датчики, перемещаемые особенности и т. д. Были разработаны дополнительно сервисные интерфейсы для обнаружения, обращения с запросом и получения ГИ, включая Web Map Server (WMS), Web Feature Service (WFS), Filter Encoding, Web Coverage Service (WCS), Catalogue Services for the Web (CSW), Web Processing Service (WPS) и др. [18]. Эти сервисные интерфейсы определяют принятую форму запросов Web именно для ГИ. Разработчики онтологий и семантической геоинтероперабельности уделили значительное внимание проблемам соответствия понятий и интеграции онтологий на основе исследования подобия между понятиями и их близости. Можно предвидеть, что в результате будет создан комплекс сервисов и приложений для увеличения возможностей вывода и взаимодействия между геоданными, полученными из многих источников Web.

8 Заключение

Семантическая геоинтероперабельность стала областью активных научных исследований геоинформатики [19]. Представленная работа носит обзорный характер и посвящена описанию составляющих, которые необходимы для развития семантической геоинтероперабельности.

Семантическую геоинтероперабельность можно сравнить с эффективным процессом двунаправленной связи, где пользователь и поставщик ГИ взаимодействуют через запросы и ответы, понимая друг друга благодаря их знаниям и процессам

рассуждения. Пространственные инфраструктуры данных для осуществления функциональной совместимости ГИ в значительной степени уже разработаны. Следующий шаг должен быть сделан в направлении реализации семантической геоинтероперабельности. Введение парадигмы Semantic Web обеспечивает новые возможности для геоинтероперабельности ГИ типа автоматической интерпретации, рассуждения и вывода.

Литература

1. *Manso-Callejo M., Wachowicz M., Bernabé-Poveda M.* Automatic metadata creation for supporting interoperability levels of spatial data infrastructures // Spatial Data Infrastructure Convergence: Building SDI Bridges to Address Global Challenges: 11th Conference (International) of the GSDI Association. Rotterdam, The Netherlands, 2009. http://www.gsdi.org/gsdiconf/gsdi11/papers/pdf/194.pdf.
2. *Kingston G., Fewell S., Richer W.* An organisational interoperability agility model. — Canberra, Australia: DSTO Fern Hill, Department of Defence, 2005. http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA463924&Location=U2&doc=GetTRDoc.pdf.
3. *Turnitsa C., Tolk A.* Battle management language: A triangle with five sides // Simulation Interoperability Standards Organization (SISO) Spring Simulation Interoperability Workshop (SIW) Proceedings. — Huntsville, AL, USA, 2006. Paper No. 06S-SIW-016.
4. *Probst F.* Ontological analysis of observations and measurements // Geographic information science / Eds. M. Raubal, H. J. Miller, A. U. Frank, M. F. Goodchild. Lecture notes in computer science ser. — Berlin–Heidelberg: Springer-Verlag, 2006. Vol. 4197. P. 304–320.
5. Geospatial service-oriented architecture (SOA). White Paper. — Redlands, CA, USA: ESRI, 2007. http://www.esri.com/library/whitepapers/pdfs/geospatial-soa.pdf.
6. *Гаврилова Т. А., Хорошевский В. Ф.* Базы знаний интеллектуальных систем. — СПб: Питер, 2000. 384 с.
7. *Cohn A. G., Gotts N. M.* The “Egg–Yolk” representation of regions with indeterminate boundaries // Geographic objects with indeterminate boundaries / Eds. P. A. Burrough, A. U. Frank. — London: Taylor & Francis, 1996. 171–187.
8. *Nonaka I., Takeuchi H.* The knowledge creating company: How Japanese companies create the dynamics of innovation. — Oxford: Oxford University Press, 1995. 295 p.
9. *Дулин С. К., Поповидченко В. Г.* Структура представления онтологии геоданных. — М.: ВЦ РАН, 2007. 23 с.
10. *Fonseca F., Egenhoter M. J., Davis Jr. C. A., Borges K. A. V.* Ontologies and knowledge sharing in urban GIS // Comput. Environ. Urban Syst., 2000. Vol. 24. No. 3. P. 251–272.
11. *Fonseca F., Egenhofer M.* Ontology-driven geographic information systems // GIS’99: 7th ACM Symposium (International) on Advances in Geographic Information Systems Proceedings. — New York, NY, USA: ACM Press, 1999. P. 14–19.
12. *Hjelmager J., Moellering H., Cooper A., Delgado T., Rajabifard A., Rapant P., Danko D., Huet M., Laurent D., Aalders H., Iwaniak A., Abad P., Düren U.,*

Martynenko A. An initial formal model for spatial data infrastructures // Int. J. Geogr. Inf. Sci., 2008. Vol. 22. P. 1295–1309.

13. ISO/IEC 10746:1998 Information technology — Open distributed processing — Reference model: Overview. — Geneva: International Organization for Standardization, 1998. http://www.techstreet.com.products/862472?product_id=862472&sid=goog&gclid=CMbwiqWBg78CFauQcgodLF4A2Q.
14. ISO 19101:2002 Geographic information — Reference model. — Geneva: International Organization for Standardization, 2002. http://www.techstreet.com/products/1031069.
15. *Berners-Lee T., Hendler J., Lassila O.* The semantic Web // Sci. Am., 2001, May. P. 34–43.
16. *Berners-Lee T., Shadbolt N., Hall W.* The semantic Web revisited // IEEE Intell. Syst., 2006. Vol. 21. No. 3. P. 96–101.
17. *Egenhofer M. J.* Toward the semantic geospatial Web // GIS'02: 10th ACM Symposium (International) on Advances in Geographic Information Systems Proceedings. — New York, NY, USA: ACM Press, 2002. P. 1–4.
18. *Brodeur J., Bédard Y., Moulin B.* A geosemantic proximity-based prototype for the interoperability of geospatial data // Comput. Environ. Urban Syst., 2005. Vol. 29. No. 6. P. 669–698.
19. *Дулин С. К., Дулина Н. Г., Кожунова О. С.* Когнитивная интероперабельность экспертной деятельности и ее приложение в геоинформатике // КИИ-2012: Труды 13-й Национальной конф. по искусственному интеллекту с международным участием. — Белгород: БГТУ им. В. Г. Шухова, 2012. Т. 1. С. 351–357.

Поступила в редакцию 24.03.14

ABOUT PROBLEMS OF IMPLEMENTATION OF SEMANTIC GEOINTEROPERABILITY IN SEMANTIC WEB

***S. K. Dulin*[1,2], *N. G. Dulina*[3], *and D. A. Nikishin*[2]**

[1]Research & Design Institute for Information Technology, Signalling and Telecommunications on Railway Transport (JSC NIIAS), 27-1 Nizhegorodskaya Str., Moscow 109029, Russian Federation

[2]Institute of Informatics Problems, Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation

[3]Dorodnicyn Computing Center, Russian Academy of Sciences, 40 Vavilov Str., Moscow 119333, Russian Federation

Abstract: The problem of semantic geointeroperability consists in supporting coordinated interaction of experts for solving tasks demanding georesources shared use, under condition of adequate understanding of semantics of the georesources. Support of semantic geointeroperability assumes that development of tools of coordinated understanding of geodata should be carried out on the basis of the comparative analysis of existing metaschemes of bases of geodata in view

of multifactor interactions of users and semantics incorporated in space ontology and/or geothesauruses and qualifiers. The key task of semantic geointeroperability is creation of a uniform conceptual model of representation and coordinated understanding of geodata on the basis of integration of the space-distributed information. The paper is devoted to discussion of the concept of semantic interoperability, aspects of implementation of semantic geointeroperability of the geographical information, and standards of semantic geointeroperability.

Keywords: geodata; semantic geointeroperability; ontology; Semantic Web

DOI: 10.14357/08696527140210

Acknowledgments

The work is supported by the Russian Foundation for Basic Research (projects 14-07-00040 and 14-07-00785).

References

1. *Manso-Callejo, M., M. Wachowicz, and M. Bernabé-Poveda.* 2009. Automatic metadata creation for supporting interoperability levels of spatial data infrastructures. *Spatial Data Infrastructure Convergence: Building SDI Bridges to Address Global Challenges: 11th Conference (International) of the GSDI Association*. Rotterdam, The Netherlands. Available at: http://www.gsdi.org/gsdiconf/gsdi11/papers/pdf/194.pdf (accessed April 4, 2014).
2. Kingston, G., S. Fewell, and W. Richer. 2005. An organisational interoperability agility model. [online]. Retrieved on October 1, 2008 from: Available at: http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA463924&Location=U2&doc=GetTRDoc.pdf (accessed March 31, 2014).
3. Turnitsa, C., and A. Tolk. 2006. Battle management language: A triangle with five sides. *Simulation Interoperability Standards Organization (SISO) Spring Simulation Interoperability Workshop (SIW) Proceedings*. Huntsville, AL, USA. Paper No. 06S-SIW-016.
4. Probst, F. 2006. Ontological analysis of observations and measurements. *Geographic information science*. Eds. M. Raubal, H. J. Miller, A. U. Frank, and M. F. Goodchild. Lecture notes in computer science ser. Berlin–Heidelberg: Springer-Verlag. 4197:304–320.
5. ESRI. 2007. Geospatial service-oriented architecture (SOA). White Paper. Redlands, CA, USA: ESRI. Retrieved on 1 October 2008 from: http://www.esri.com/library/whitepapers/pdfs/geospatial-soa.pdf (accessed March 31, 2014).
6. Gavrilova, T. A., and V. F. Khoroshevskij. 2000. *Bazy znaniy intellektual'nykh system* [*Knowledge base for intelligent systems*]. St. Petersburg: Piter. 384 p.
7. Cohn, A. G., and N. M. Gotts. 1996. The "Egg–Yolk" representation of regions with indeterminate boundaries. Eds. P. A. Burrough and A. V. Frank. *Geographic objects with indeterminate boundaries*. London: Taylor & Francis. 171–187.

8. Nonaka, I., and H. Takeuchi. 1995. *The knowledge creating company: How Japanese companies create the dynamics of innovation*. Oxford: Oxford University Press, 1995. 295 p.
9. Dulin, S. K., and V. G. Popovidchenko. 2007. *Struktura predstavleniya ontologii geodannykh* [*Structure ontology representation of geodata*]. Moscow: Dorodnicyn Computing Center of RAS. 23 p.
10. Fonseca F., M. J. Egenhoter, C. A. Davis, Jr., and K. A. V. Borges. 2000. Ontologies and knowledge sharing in urban GIS. *Comput. Environ. Urban Syst.* 24(3):251–272.
11. Fonseca, F., and M. Egenhofer. 1999. Ontology-driven geographic information systems. *7th ACM Symposium on Advances in Geographic Information Systems*. New York, NY, USA: ACM Press. 14–19.
12. Hjelmager, J., H. Moellering, A. Cooper, T. Delgado, A. Rajabifard, P. Rapant, D. Danko, M. Huet, D. Laurent, H. Aalders, A. Iwaniak, P. Abad, U. Düren, and A. Martynenko. 2008. An initial formal model for spatial data infrastructures. *Int. J. Geogr. Inf. Sci.* 22:1295–1309.
13. ISO/IEC 10746:1998 Information technology — Open distributed processing — Reference model: Overview. 1998. International Organization for Standardization. Geneva. Available at: http://www.techstreet.com.products/862472?product_id=862472&sid=goog&gclid=CMbwiqWBg78CFauQcgodLF4A2Q (accessed June 24, 2014).
14. International Organization for Standardization. 2002. ISO 19101:2002 Geographic information — Reference model. Geneva. Available at: http://www.techstreet.com/products/1031069 (accessed June 24, 2014).
15. Berners-Lee, T., J. Hendler, and O. Lassila. 2001. The semantic Web. *Sci. Am.* May:34–43.
16. *Berners-Lee, T., N. Shadbolt, and W. Hall*. 2006. The Semantic Web revisited. *IEEE Intell. Syst.* 21(3):96–101.
17. Egenhofer, M. J. 2002. Toward the semantic geospatial Web. *GIS'02: 10th ACM Symposium (International) on Advancies in Geographic Information Systems Proceedings*. New York, NY, USA: ACM. 1–4.
18. Brodeur, J., Y. Bédard, and B. Moulin. 2005. A geosemantic proximity-based prototype for the interoperability of geospatial data. *Comput. Environ. Urban Syst.* 29:669–698.
19. Dulin, S. K., N. G. Dulina, and O. S. Kozhunova. 2012. Kognitivnaya interoperabel'nost' ekspertnoy deyatel'nosti i ee prilozhenie v geoinformatike [Cognitive interoperability expert activity and its application in geoinformatics]. *Trudy 13-j Natsional'noy Konferentsii po Iskusstvennomu Intellektu "KII-2012"* [*13th National Conference on Artificial Intelligence "CAI-2012" Proceedings*]. Belgorod. 1:351–357.

Received March 24, 2014

Contributors

Dulin Sergey K. (b. 1950) — Doctor of Science in technology, professor; principal scientist, Research & Design Institute for Information Technology, Signalling and

Telecommunications on Railway Transport (JSC NIIAS), 27-1 Nizhegorodskaya Str., Moscow 109029, Russian Federation; senior scientist, Institute of Informatics Problems, Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; s.dulin@ccas.ru

Dulina Natalia G. (b. 1947) — Candidate of Science (PhD) in technology, senior scientist, Dorodnicyn Computing Center, Russian Academy of Sciences, 40 Vavilov Str., Moscow 119333, Russian Federation; ngdulina@mail.ru

Nikishin Dmitry A. (b. 1976) — Candidate of Science (PhD) in technology, Head of Laboratory, Institute of Informatics Problems, Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; dmnikishin@mail.ru

АНАЛИТИЧЕСКИЕ АСПЕКТЫ МУЛЬТИАГЕНТНЫХ РАСПРЕДЕЛЕННЫХ СИСТЕМ УПРАВЛЕНИЯ

А. П. Сучков[1]

Аннотация: Рассматриваются вопросы применения мультиагентного подхода к созданию интегрированных систем ситуационного управления с учетом их характерных особенностей, таких как сложность среды взаимодействия, независимость, автономность, децентрализация. Изучается организация мультиагентной системы управления и структура агента, реализующего модифицированный цикл управления НОРД (наблюдение–ориентирование–решение–действие) для создания системы распределенных ситуационных центров (СРСЦ). Систематизированы методы анализа данных на всех стадиях цикла управления. Изучается структура среды взаимодействия распределенной системы управления, обеспечивающей обмен не только событийной информацией, но и данными аналитики с целью поддержки процессов принятия решений при реализации целей управляющей системы. Разработаны способы формализации аналитических данных с учетом всех составляющих контролируемого пространства распределенной системы управления, включая целевую обстановку, контролируемые объекты, контролируемые ресурсы, неконтролируемые факторы.

Ключевые слова: мультиагентная система; ситуационный центр; распределенная система управления; аналитические данные

DOI: 10.14357/08696527140211

1 Введение

Статья посвящена особенностям организации асинхронных процессов информационного взаимодействия в сложных сетецентрических системах управления. Мультиагентный подход к созданию сложных сетевых систем управления может быть актуален при реализации Указа Президента РФ «О формировании системы распределенных ситуационных центров, работающих по единому регламенту взаимодействия» и интеграции других разнородных автоматизированных ведомственных систем управления.

Целью создания СРСЦ является обеспечение информационно-аналитической поддержки государственного управления, стратегического планирования и мониторинга реализации документов стратегического планирования в Российской

[1]Институт проблем информатики Российской академии наук, asuchkov@ipiran.ru

Федерации, повышение эффективности государственного управления в повседневном режиме, а также при возникновении кризисных и/или чрезвычайных ситуаций на основе информационных и технологических возможностей ситуационных центров (СЦ) органов государственной власти (ОГВ), обеспечивающих анализ, оценку, прогнозирование изменения обстановки и поддержку принятия управленческих решений.

Для достижения указанной цели в числе прочих предусматривается обеспечить выполнение следующих задач:

- создание новых и модернизация действующих СЦ в ОГВ Российской Федерации с применением перспективных типовых решений по составу программно-технических комплексов, обеспечивающих эффективное информационное и технологическое взаимодействие в СРСЦ, а также повышение качества технического оснащения ОГВ;
- формирование территориально-распределенного информационного фонда СРСЦ, обеспечивающего доступ к информационным ресурсам СЦ ОГВ на основе портальной технологии;
- организация управления СРСЦ и координации взаимодействия СЦ ОГВ в системе на основе единого регламента.

Кроме того, в качестве одного из основных принципов создания СРСЦ выдвигается принцип матричного взаимодействия, обеспечивающий одновременное использование двух типов взаимодействия СЦ: иерархический и сетевой. Иерархический тип используется для взаимодействия СЦ в рамках вертикалей ведомственной, территориальной и отраслевой подчиненности. Сетевое взаимодействие осуществляется в рамках обмена информацией тематической направленности с любыми СЦ в режиме оперативных совещаний, в том числе при возникновении чрезвычайных ситуаций.

Таким образом, СРСЦ характеризуется как распределенная сетецентрическая информационно-аналитическая система ситуационного управления. Рассмотрим подходы к созданию СРСЦ с использованием технологий мультиагентных систем и особенности их реализации с точки зрения распределенных систем поддержки принятия решений.

2 Мультиагентная распределенная система управления

Под мультиагентной системой управления понимается система, образованная несколькими интеллектуальными агентами, взаимодействующими между собой и с внешней средой. В связи с этим в мультиагентной системе организуется среда взаимодействия, обеспечивающая обмен информацией между отдельными агентами (например, сообщениями о событиях, запросами, инструкциями), а также определенным образом структурируется информация о внешней среде (рис. 1).

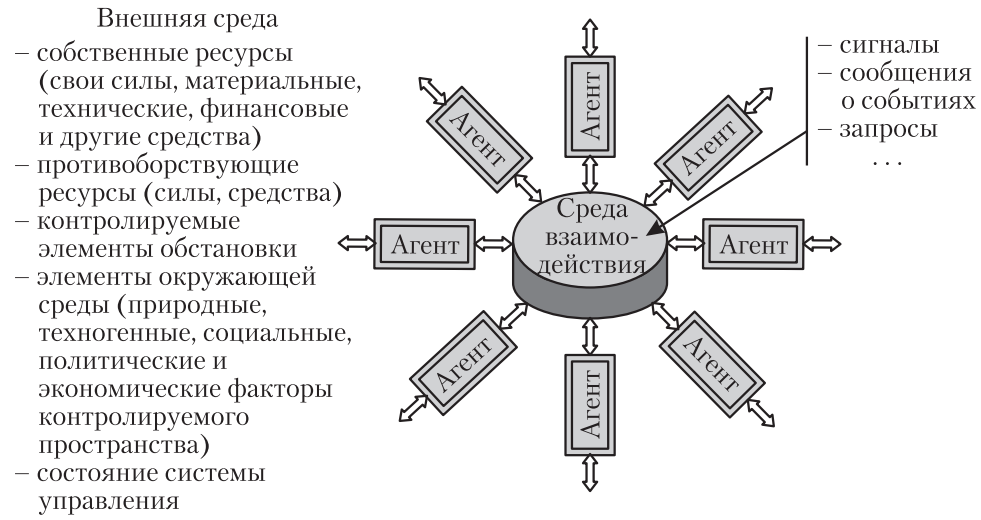


Рис. 1 Обобщенная структура мультиагентной системы

Интеллектуальный агент — сущность, получающая информацию через систему сенсоров о состоянии управляемых ею процессов, анализирующая ее и осуществляющая целенаправленное влияние на них через систему актуаторов.

В мультиагентной системе агенты независимы хотя бы частично, обладают способностью к самоорганизации, т. е., располагая собственными ресурсами, способны адаптироваться к изменениям внешней среды, а также децентрализованы, т. е. нет агентов, управляющих всей системой.

Для того чтобы конкретизировать структуру интеллектуального агента, реализующего функции управления, можно использовать одну из адекватных моделей управленческого цикла для принятия решений в режиме реального времени — модель OODA (НОРД). В конце 1970-х гг. Бойд создал данную модель для принятия решений при ведении боевых действий [1]. В настоящее время эта модель активно используется во многих системах управления в условиях конкурентной среды. Основные элементы цикла:

– **наблюдение** (*observation*) — это процесс сбора информации как от внешних, так и от внутренних источников, необходимой для принятия решения;

– **ориентирование** (*orientation*) — этап ориентирования состоит в определении того, является ли ситуация известной (типовой) или совокупностью типовых ситуаций, для которых у лица, принимающего решение (ЛПР), имеется план решения;

– **решение** (*decision*) — ЛПР на данном этапе осуществляет выбор наилучшего из альтернативных вариантов действий для последующей реализации;

– **действие** (*action*) — реализация избранного плана действий и контроль исполнения путем перехода к этапу **наблюдение**.

Каждая стадия цикла управления, связанная с процессом принятия решений, опирается на совокупность методов анализа данных, обеспечивающую интеллектуальную поддержку их функциональности.

Стадия **наблюдение** использует методы анализа сигналов, фактов и формализации данных:

– анализ сигналов, распознавание образов, оценка ненаблюдаемых элементов обстановки;

– анализ текстов на естественном языке и слабоструктурированных данных с целью выделения из них контролируемых информационных объектов предметной области и их взаимосвязей;

– формирование семантической сети событийной информации на основе интеграции всех поступающих данных об обстановке (идентификация и регистрация объектов и установление их взаимосвязей);

– автоматизированный учет, рубрицирование, реферирование и перевод документов.

Стадия **ориентирование** использует методы анализа взаимосвязей на дискретной модели, а также анализ массовых потоков событий:

– поиск неочевидных взаимосвязей (путей на графе);

– поиск схожих пространственно-временны́х конфигураций событий (идентификация ситуаций);

– визуализация и навигация по семантической сети;

– формирование выборок, рядов и OLAP (online analytic processing) кубов;

– экспресс-анализ ситуаций;

– динамическое моделирование ситуаций;

– прогнозирование ситуаций, сценарное прогнозирование;

– анализ временны́х рядов;

– статистическая оценка характеристик потоков событий.

Стадии **решение** и **действие** опираются на комплекс информационно-расчетных и информационно-аналитических задач, связанных с планированием применения сил и средств:

– картометрические задачи (геомоделирование и геопрогнозирование);

– тематический анализ зон;

– расчеты зон достижимости и оптимальных маршрутов;

– расчеты вероятностей выполнения задач.

Грант в 2005 г. предложил модель интеллектуального агента на основе **модифицированного** цикла НОРД [2]. Особенностью этой модели является учет наличия штатных и нештатных ситуаций и целесообразности функционирования системы управления. Он ввел в состав агента базу данных (БД) типовых ситуаций и планов их нормализации, БД текущих целей управления, стадию **решение** он разбил на три составляющих: планирование применения сил и средств по типовым ситуациям, выработка планов решений по нештатным ситуациям группой экспертов и непосредственно принятие решения в увязке с текущими целями управления. При этом в среду взаимодействия таких агентов он включил не только сигналы внешней среды и регистрируемые события, но и инструкции — указания или запросы на решение управленческой задачи.

В 2008 г. в работе Ивлева [3] рассмотрена возможность создания **сетевой структуры управления** на основе цикла НОРД, где каждая стадия реализуется, по сути, в виде самостоятельного агента НОРД, при этом структура среды взаимодействия дополняется такими элементами, как «ситуация» и «план».

Таким образом, рассматривая мультиагентную систему управления, можно отметить следующее: внешняя среда разбита на зоны ответственности агентов, в которых фиксируются события, характеризующие изменения состояния контролируемых объектов. События группируются в ситуации, при этом штатные ситуации нормализуются силами ответственного агента, а нештатные — в основном выходящие за пределы одной зоны ответственности — передаются в среду взаимодействия мультиагентной системы для ее нормализации силами и средствами нескольких агентов. Для обеспечения возможности принятия совместных решений необходим также обмен формализованными и неформализованными **аналитическими данными**, которые также становятся доступными в среде взаимодействия. Обмениваться результатами аналитики может быть выгодно, так как ресурсоемкие продукты, связанные с оценками значений ненаблюдаемых параметров внешней среды, с прогнозом развития характеристик элементов обстановки, связанные с выявленными закономерностями и схожими пространственно-временны́ми конфигурациями событий, являются во многом уникальными и их многоаспектное использование повышает эффективность всей системы управления.

3 Состав среды взаимодействия системы распределенных ситуационных центров

Система распределенных ситуационных центров призвана объединить уже созданные и вновь создаваемые СЦ центральных органов власти, федеральных ведомств, полномочных представителей Президента РФ в федеральных округах, глав субъектов РФ числом более 70. В составе СРСЦ должен быть создан **Центр управления и координации**, осуществляющий реализацию единого регламента взаимодействии, включая (рис. 2):

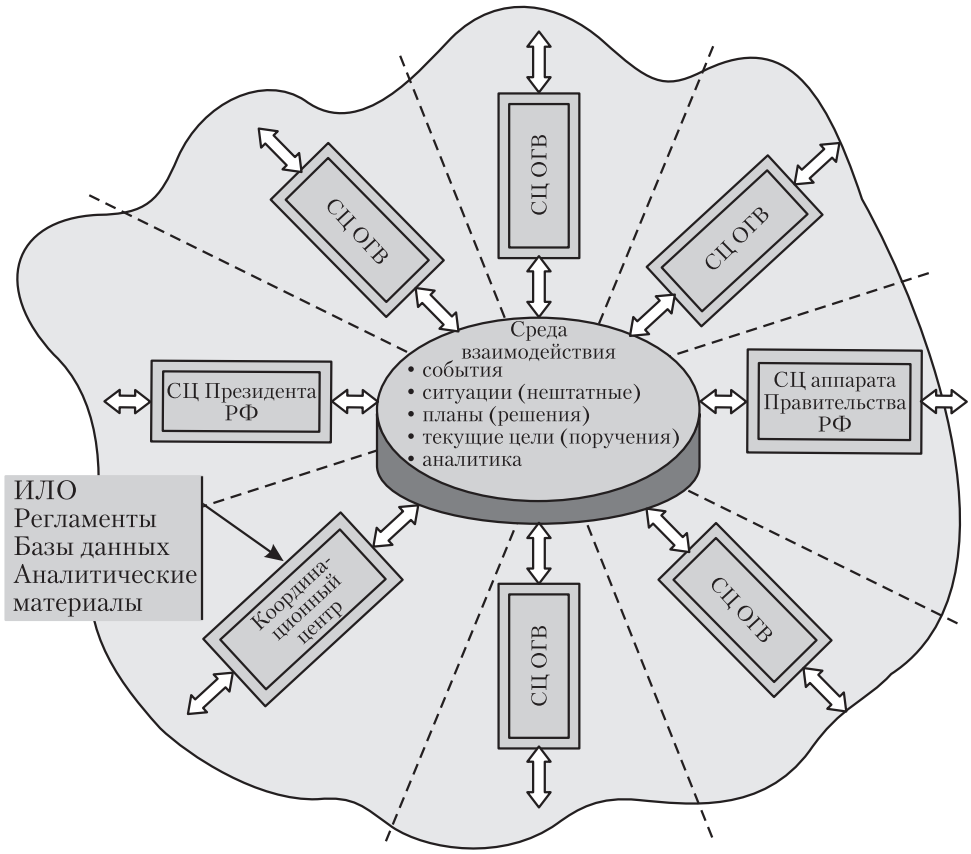


Рис. 2 Структура СРСЦ

– единое информационно-лингвистическое обеспечение (ИЛО) СРСЦ;
– единые регламенты взаимодействия, реализующие среду взаимодействия мультиагентной системы в части:
  - событийной информации;
  - распознанных и нераспознанных ситуаций;
  - инструкций (планов, целей, задач);
– ведение вспомогательных БД:
  - событий;
  - типовых ситуаций (типовых решений);
  - текущих целей системы управления;
– аналитические материалы.

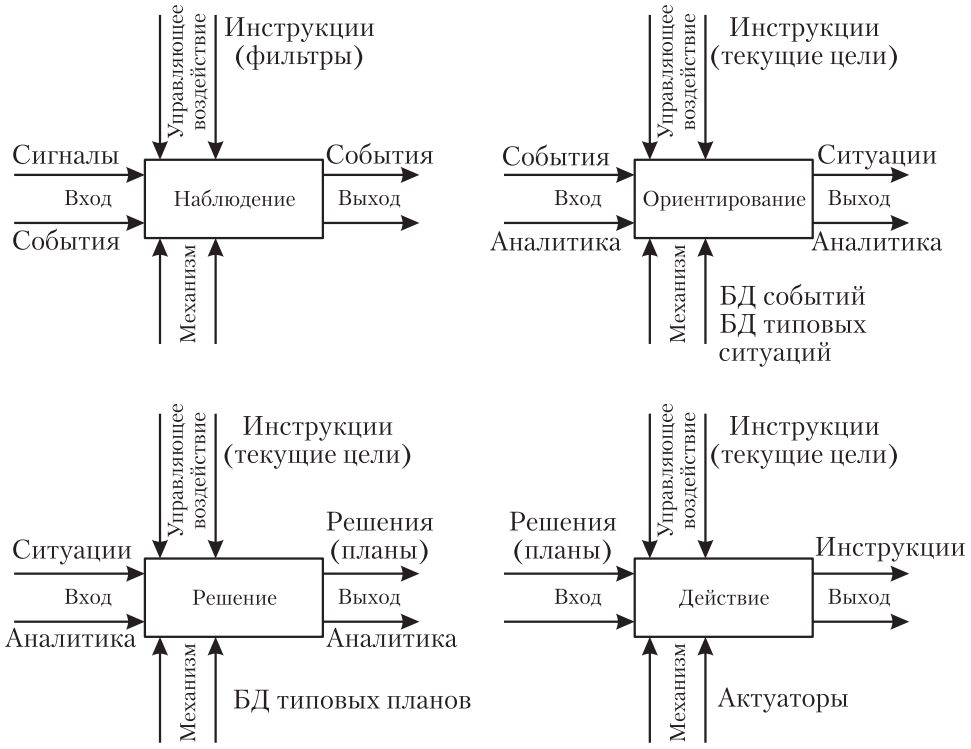


Рис. 3 Состав и структура среды взаимодействия СРСЦ

В составе каждого СЦ необходимо на основе типовых решений создать агенты, реализующие в среде взаимодействия асинхронные обмены данными в рамках реализации цикла управления НОРД (рис. 3). Функционал таких агентов достаточно прост — они должны обеспечивать трансляцию соответствующих регламентированных данных из СЦ в среду взаимодействия и в обратном направлении.

4 Формализация аналитических данных

Рассмотрим способы формализации аналитических данных в среде взаимодействия мультиагентной системы ситуационного управления с точки зрения поддержки в СРСЦ процессов государственного стратегического планирования. Эти процессы обусловлены решением задач управления, связанных с анализом массовых потоков событий по всем направлениям деятельности государства — политическому, экономическому, социальному, обеспечения национальной без-

опасности. Для анализа обстановки и поддержки процессов принятия решений на этих уровнях применяются математические методы статистического анализа потоков событий с целью создания динамических моделей процессов. На их основе осуществляется анализ текущего состояния, сравнительный анализ с прошедшим периодом времени, выявление тенденций и аномалий в потоках событий, прогнозирование развития ситуаций и угроз, поддержка принятия решений.

Во-первых, применение математических методов анализа предполагает оцифровку изучаемых процессов. Ситуационный анализ, реализующий процессы поддержки принятия решений в системах управления, опирается на такие понятия, как событие, обстановка, ситуация, управление. Ситуация определяется состоянием взаимосвязанных элементов обстановки в контролируемом пространстве, изменения состояния элементов обстановки определяются событиями. При этом под управлением понимается целенаправленное воздействие органа управления на подчиненные ему или взаимодействующие элементы обстановки (ресурсы) [4, 5].

С целью формализации аналитических данных обобщенную обстановку (внешнюю среду) СЦ можно структурировать следующим образом:

- целевая обстановка (совокупность плановых состояний обстановки);
- контролируемые объекты (объекты, состояние которых подлежит контролю с точки зрения целей управления);
- контролируемые ресурсы (объекты управления, например свои или взаимодействующие ресурсы);
- неконтролируемые факторы, например противоборствующие ресурсы (силы, средства), элементы окружающей среды (природные, техногенные, социальные, политические и экономические факторы контролируемого пространства).

Основными показателями динамики изменения обстановки являются **интегральные показатели**, связанные с потоками происходящих событий [5]. Например, простейший показатель — количество событий определенного типа, в определенном месте и за определенное время или суммарная числовая характеристика элемента обстановки (например, площадь лесных пожаров). Цели управления определяются **целевыми показателями** и их плановыми значениями. Целевой показатель можно определить как функцию от набора интегральных показателей обстановки, например это может быть взвешенная сумма таких показателей. Для каждой предметной области СРСЦ необходимо сформировать наборы интегральных и целевых показателей и организовать сбор данных о событиях в зонах ответственности, позволяющий получить наборы данных оцифрованной обстановки.

Во-вторых, организованные в виде временны́х рядов данные об обстановке позволяют применять различные виды статистического анализа, состав которых обсужден выше. Можно выделить пять стадий ситуационного анализа, основанных на подходе НОРД:

(1) оценка параметров ненаблюдаемых (скрытых) элементов обстановки на основе выборочных или косвенных данных по результатам мониторинга;

(2) оперативный анализ обстановки путем ее сравнения с прошедшим периодом (без изменений, хуже, лучше, аномалия) с целью выявления ситуаций, требующих немедленного реагирования;

(3) оценка ситуации с целью определения необходимости выработки решений по ее нормализации и степени сложности ситуации — штатная, критическая, чрезвычайная;

(4) прогнозирование изменения обстановки — без управляющего воздействия, с управляющим воздействием, сценарное прогнозирование с учетом внешних факторов;

(5) поддержка принятия управленческих решений — адаптация типовых решений и выработка нетиповых решений (с учетом прогнозирования).

Состав и группировка элементов среды взаимодействия

Обстановка: виды анализа	Обстановка: Элементы среды взаимодействия СРСЦ (примеры)			
	Целевая обстановка	Контролируемые объекты	Контролируемые ресурсы	Неконтролируемые факторы
Оцифровка обстановки	Целевые показатели	Интегральные показатели	Интегральные показатели	Интегральные показатели
Оценка скрытых элементов обстановки		Оценка параметра	Оценка параметра	Оценка параметра
Оперативный анализ обстановки	Ситуации	Ситуации	Ситуации	Ситуации
Оценка ситуации	Оценка	Оценка	Оценка	Оценка
Прогнозирование изменения обстановки	Прогноз значения целевых показателей	Прогноз значения интегральных показателей	Прогноз значения интегральных показателей	Прогноз значения интегральных показателей
Поддержка принятия управленческих решений	Сценарный прогноз. Решение	Сценарный прогноз	Сценарный прогноз. Инструкции	Сценарный прогноз

В соответствии с приведенной структуризацией обстановки и стадиями ситуационного анализа можно определить ориентировочный состав элементов среды взаимодействия мультиагентной СРСЦ и их группировку в зависимости от вида решаемой управленческой задачи (см. таблицу).

Таким образом, можно предположить, что для взаимодействия в рамках СРСЦ в плане аналитики достаточно применение следующих кортежей данных:

– для значений показателей: ⟨*источник, дата, тип параметра (целевой, интегральный), название параметра, значение, единица измерения*⟩;

– для оценки параметра: ⟨*источник, дата, название параметра, значение, единица измерения, доверительный интервал, вероятность*⟩;
– для идентификации и оценки ситуаций: ⟨*источник, дата, тип ситуации (штатная, чрезвычайная, критическая), вид ситуации (по классификатору), локализация (по классификатору)*⟩;
– для обмена прогнозными данными: ⟨*источник, дата, тип параметра (целевой, интегральный), название параметра, временной вектор, вектор значений, единица измерения, доверительный интервал, вероятность*⟩;
– для обмена данными по сценарному прогнозированию: ⟨*источник, дата, тип параметра (целевой, интегральный), название параметра, сценарий (по классификатору), временной вектор, вектор значений, единица измерения, доверительный интервал, вероятность*⟩;
– для передачи данных по решениям: ⟨*источник, дата, ситуация (регистрационный №), действие (по классификатору), исполнитель, срок исполнения, название целевого параметра, плановое значение, единица измерения*⟩.

Как видно, приведенная формализация аналитических данных должна опираться на единую систему классификации и единую систему учета основных элементов обстановки и событийной информации. Среда взаимодействия должна также поддерживать обмены неформализованными данными, обеспечивая тем самым полноту многообразной картины ситуационных данных.

5 Выводы

1. Создание сетевых мультиагентных систем управления является перспективным направлением, особенно в задачах интеграции уже существующих многочисленных автоматизированных ведомственных и корпоративных систем управления.
2. При разработке таких интегрированных систем, как СРСЦ, целесообразно использовать модифицированную модель цикла управления НОРД в его агентной реализации.
3. Для создания мультиагентной системы из уже существующих автоматизированных систем управления необходимо сформировать среду взаимодействия и наборы агентов — трансляторов данных.
4. В состав среды взаимодействия должны включаться формализованные и неформализованные аналитические данные.

Литература

1. *Boyd J. R.* Patterns of conflict. Unpublished briefing slides, 1986.
2. *Grant T.* Unifying planning and control using an OODA-based architecture // SAICSIT Proceedings, 2005. P. 111–113. http://www.uran.donetsk.ua/~masters/2012/fknt/bakshevnikova/library/grant.pdf.

3. *Ивлев А. А.* Основы теории Бойда. Направления развития, применения и реализации. — М., 2008. 64 с. http://www.slideshare.net/defensenetwork/ss-10380168.
4. *Сучков А. П.* Два подхода к ситуационному анализу потоков событий // Ситуационные центры: фокус кросс-отраслевых интересов: Мат-лы II конф. Москва: РАНХиГС при Президенте РФ. http://www.ситцентр.рф/archive/2012/obzor.html.
5. *Зацаринный А. А., Сучков А. П.* Некоторые подходы к ситуационному анализу потоков событий // Открытое образование, 2012. № 1. С. 39–45.

Поступила в редакцию 18.02.14

ANALYTICAL ASPECTS OF MULTIAGENT DISTRIBUTED CONTROL SYSTEMS

A. P. Suchkov

Institute of Informatics Problems, Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation

Abstract: The paper considers the issues of application of the multiagent approach for creation of integrated systems of situational management, taking into account their specific features: complexity of environment interaction, independence, autonomy, and decentralization. The paper studies organization of multiagent control systems and agent framework, implements a modified management cycle OODA (observation–orientation–decision–action) to create a system of distributed situational centers. The methods of data analysis used at all stages of the management cycle were systematized. The structure of a collaborative environment for distributed control system was studied. The system exchanges not only event information, but also data analytics to support decision making while realizing the objectives of the control system. The ways of formalizing the analytical data were developed with regard to all components of the control area of the distributed control system, including targeted environment, controlled objects, controlled resources, and uncontrolled factors.

Keywords: multiagent system; situational center; distributed control system; analytical data

DOI: 10.14357/08696527140211

References

1. Boyd, J. R. 1986. Patterns of conflict. Unpublished briefing slides.
2. Grant, T. 2005. Unifying planning and control using an OODA-based architecture. *SAICSIT Proceedings*. 111–113. Available at: http://www.uran.donetsk.ua/~masters/2012/fknt/bakshevnikova/library/grant.pdf (accessed June 24, 2014).

3. Ivlev A. A. 2008. Osnovy teorii Boyda. Napravleniya razvitiya, primeneniya i realizatsii [Fundamentals of the theory of Boyd. Directions of development, application and implementation]. Available at: http://www.slideshare.net/defensenetwork/ss- 10380168 (accessed March 24, 2014).
4. Suchkov, A. P. 2012. Dva podkhoda k situatsionnomu analizu potokov sobytiy [Two approaches to the situational analysis of the flows of events]. *2nd Conference "The Situational Centers: The Focus of Cross-Sectoral Interests." RANHiGS*. Available at: http://www.ситцентр.рф/archive/2012/obzor.html (accessed March 24, 2014).
5. Zatsarinny, A. A., and A. P. Suchkov. 2012. Nekotorye podkhody k situatsionnomu analizu potokov sobytiy [Some approaches to the situational analysis of the flows of events]. *Otkrytoe Obrazovanie* [*Open Education*] 1:39–45.

Received February 18, 2014

Contributor

Suchkov Alexander P. (b. 1954) — Doctor of Science in technology, leading scientist, Institute of Informatics Problems, Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; Asuchkov@ipiran.ru

СРЕДСТВА ПОДДЕРЖКИ ИНТЕРНЕТ-ПОИСКА ПРИ ПРОВЕДЕНИИ БИОГРАФИЧЕСКИХ ИССЛЕДОВАНИЙ

И. М. Адамович[1], *О. И. Волков*[2]

Аннотация: Рассматривается значимая часть биографического исследования (БИ) — поиск информации в сети Интернет. Исследуются причины трудоемкости этого поиска. С этой целью в работе рассматриваются виды информационных потребностей пользователя при проведении БИ. Выделены специфические для БИ информационные потребности и описан связанный с ними вид поиска — «косвенный поиск». Сформулированы требования к технологии интернет-поиска, поддерживающей косвенный поиск и призванной существенно упростить его проведение. Описан и проанализирован пример такой технологии, организованной в форме набора правил без привлечения специального программного обеспечения (ПО). На основании анализа недостатков этой технологии сформулированы требования к специальному ПО поддержки поиска. Описана технология, базирующаяся на таком ПО, свободная от выявленных недостатков, позволяющая осуществлять эффективный биографический поиск и поддерживающая хранение его результатов и доступ к ним. Данная технология предназначена для широкого круга не являющихся профессиональными историками и биографами пользователей, что актуально в связи со все увеличивающимся общественным интересом к семейной истории.

Ключевые слова: биографическое исследование; технология интернет-поиска

DOI: 10.14357/08696527140212

1 Введение

Как правило, в работах, посвященных методологическим вопросам БИ [1–5] недостаточно внимания уделяется организации интернет-поиска, хотя он является значимой частью исследования, что напрямую вытекает из специфики БИ, подробно разобранной с этой точки зрения в разд. 2 настоящей статьи.

При этом, как показывает опыт самостоятельных исследований, несмотря на кажущуюся доступность в сети Интернет больших объемов информации по самым разнообразным темам, биографический интернет-поиск (БИП) является очень трудоемким и отнимает у исследователя много сил и времени.

Целью настоящего исследования является построение и обоснование новой технологии поиска, позволяющей эффективно осуществлять БИП. Для этого:

[1]Институт проблем информатики Российской академии наук, Adam@amsd.com
[2]Институт проблем информатики Российской академии наук, Volkov@amsd.com

– были проанализированы причины, затрудняющие БИП. С этой целью в разд. 3 была произведена классификация видов информационных потребностей пользователя при проведении БИ. Была выделена специфическая для БИ информационная потребность и связанный с ней вид поиска (косвенный интернет-поиск — КИП), подробно проанализированный в разд. 4;
– на основании результатов анализа специфики КИП в разд. 5 были сформулированы требования к технологии, поддерживающей данный вид поиска. На основании этих требований и в качестве обобщения приемов поиска, выработанных при проведении реальных исследований, предложена и описана в разд. 6 технология поиска, организованная в виде набора правил (технология НП) без привлечения специального ПО;
– проведен анализ предложенной технологии в разд. 7. Выявлены недостатки описанного подхода и сделан вывод о необходимости модификации технологии и создания специальных программных средств ее поддержки. Требования к ним описаны в разд. 8;
– предложена и описана в разд. 9 технология биографического поиска, свободная от недостатков технологии НП, позволяющая осуществлять эффективный БИП, а также поддерживающая хранение результатов поиска и доступ к ним.

Научная новизна работы состоит:

– в новом подходе к классификации видов поиска исходя из видов информационной потребности, специфической для БИ, в отличие от традиционных подходов к классификации поисковых запросов либо по формальным признакам [6], либо по видам информационных услуг [7];
– в создании нового метода решения задачи БИП, основывающегося на оригинальной обоснованной технологии.

Актуальность и значимость результатов исследования вытекает из существенного роста в последнее время интереса к БИ, в том числе со стороны непрофессиональных исследователей [8], с одной стороны, и отсутствия устоявшейся и обоснованной методологии этих исследований [9], с другой стороны. Сокращению этого разрыва между возросшей потребностью и отсутствием достаточной методологической поддержки и соответствующего инструментария должна служить разработанная в рамках данного исследования технология и создаваемое для ее поддержки специализированное ПО.

2 Специфика биографических исследований

Специфика БИ состоит в том, что в центре внимания исследователя находится конкретная личность и все без исключения стороны (социальные, экономические, политические, этнические, художественные и т. п.) ее реальной жизни [10]. Многообразие изучаемых аспектов жизни индивидуума, стоящего в центре исследования, приводит к огромному числу направлений поиска интересующей биографа

информации. Ситуация для исследователя, как правило, усугубляется уникальностью обстоятельств жизни изучаемого индивидуума и фрагментарностью, несистематизированностью и низкой достоверностью исходной информации о нем [11]. Следствием этого является необходимость использования абсолютно всех доступных источников информации. Среди них Интернет, наряду с архивами и специальной литературой, является одним из наиболее значимых.

3 Виды информационных потребностей исследователя при проведении биографических исследований

Как было показано выше, интернет-поиск является важной частью БИ. Сейчас прикладываются большие усилия по повышению эффективности поиска информации в сети Интернет, в том числе увеличение мощности языка запросов и повышение эффективности механизмов ранжирования [12]. Но по опыту проведения реальных БИ можно констатировать, что этот поиск является весьма трудоемким и требует значительных временны́х ресурсов. Объясняется это тем, что существующие технологии поддержки поиска в сети Интернет ориентированы на ситуацию, когда информационная потребность пользователя в принципе может быть выражена в интернет-запросе, т. е. пертинентность (субъективная мера соответствия информации, полученной в результате поиска, потребности пользователя) воспринимается как сочетание адекватности запроса (соответствие между информационной потребностью пользователя и формализованным запросом) и релевантности (мера соответствия списка результатов запроса самому запросу) [13]. Этот подход ориентирован на массового интернет-пользователя, который осуществляет поиск, как правило, в повседневных, бытовых целях. Такой поиск, когда информационная потребность выражается в формальном запросе, будем называть прямым интернет-поиском (ПИП). К сожалению, биографический поиск в сети Интернет часто не сводится к ПИП. Специфика БИ часто порождает два вида информационной потребности, которые не могут быть разумным образом выражены в одном интернет-запросе:

(1) слишком общая потребность;

(2) слишком частная потребность.

Слишком общая информационная потребность возникает, когда вопрос, который исследователь ставит перед собой в рамках исследования, пока еще недостаточно конкретизирован, чтобы быть сформулированным для поиска в сети Интернет. Более того, БИ собственно и начинается с самого общего вопроса: «Какова биография данной персоны?» — и состоит в его последовательной конкретизации. Причем конкретизация становится возможна только после нахождения какой-то информации по более общему вопросу. Так, переход от более общего вопроса «Какова биография данного человека?» к более частному «Когда и где данный человек проходил воинскую службу?» возможна только в случае, если, пытаясь найти ответ на общий вопрос о биографии, исследователь нашел

факты, позволяющие хотя бы предположить, что факт прохождения воинской службы имел место. Таким образом, исследователь, проводя биографические изыскания, постоянно вынужден искать ответы на вопросы слишком общие, чтобы предполагать наличие в сети Интернет прямых ответов на них, естественно, если речь не идет о биографии очень известной персоны. Так, на вопрос «Биография А. С. Пушкина» можно найти прямой ответ в Интернете, но это достаточно редкая ситуация, возникающая только тогда, когда объект исследования был как-то связан с известной персоной, для которой подробное БИ уже было произведено и его результаты представляют интерес для достаточно широкого круга лиц.

Слишком частная информационная потребность возникает, когда вопрос достаточно конкретизирован, но исследователю не следует ожидать наличия в сети Интернет прямого ответа на него в силу его слишком частного характера. Например, вопрос «Где находился дом подольского мещанина такого-то?» абсолютно конкретен, но только если данный мещанин не был местной знаменитостью, ответ на него в готовом виде никак не может содержаться в сети Интернет просто потому, что ранее этот вопрос никого не интересовал.

Таким образом, исследователь, как правило, не может рассчитывать, что прямые вопросы, стоящие перед ним на данном этапе исследования, могут быть непосредственно преобразованы в некоторые интернет-запросы, на которые поисковые машины (ПМ) выдадут информацию, содержащую столь же прямые ответы на них. Он будет вынужден, опираясь на уже известную ему, как правило, фрагментарную, противоречивую и недостоверную информацию по теме исследования, придумывать и формулировать косвенные вопросы, постепенно заполняя информационные лакуны. Такой интернет-поиск будем называть косвенным интернет-поиском.

4 Специфика косвенного интернет-поиска

При проведении КИП исследователю приходится многократно переформулировать запрос, пока результаты поиска его не удовлетворят, т. е. осуществлять итерационный поиск. Анализ результатов, найденных на предыдущем шаге итерации, уточняет направление поиска на последующем шаге. Таких уточнений на одном шаге может быть несколько, поэтому КИП является не только итерационным, но и ветвящимся процессом.

На каждом шаге итерации интересующий исследователя результат с большой вероятностью будет находиться отнюдь не в первых строчках выдачи. Это объясняется тем, что современные алгоритмы ранжирования результатов выдачи учитывают авторитетность страницы, которую они вычисляют по количеству и качеству ссылок на нее с других сайтов [7]. Интерес же для исследователя часто представляет малоизвестная информация.

Также отличительной чертой КИП является то, что исследователь часто во время поиска не может быть абсолютно уверен, содержится ли в полной

мере интересующая его информация в уже найденных документах или еще нет, поскольку ответ на этот вопрос требует тщательного изучения этих документов. Поэтому КИП, как правило, проводится до тех пор, пока новые документы на интересующую тематику не перестанут появляться в выдаче или пока найденных документов не наберется достаточно много, т. е. поиск новых начнет приводить к неоправданному увеличению их количества до объемов, затруднительных для изучения в разумное время.

Таким образом, специфика КИП состоит в том, что он:

- производится, как правило, итерационно с ветвлениями;
- требует на каждой итерации просмотра значительного объема результатов выдачи;
- осуществляется «до упора», а не до нахождения первой удовлетворяющей исследователя ссылки.

5 Требования к технологии интернет-поиска при проведении биографических исследований

Эффективный поиск не может быть бессистемным и обязательно требует следования некоторым правилам, представляющим собой некоторую технологию БИП. Требования к технологии БИП вытекают из специфики КИП и необходимости сохранения и систематизации найденной информации в интересах ведущегося исследования, т. е. технология БИП должна поддерживать:

- итерационный характер поиска;
- возможность ветвлений поиска;
- просмотр значительного объема результатов выдачи;
- хранение найденной информации;
- систематизацию найденной информации.

К сожалению, программных средств, в полной мере поддерживающих данные требования, нет, хотя существуют средства систематизации и хранения биографической информации [14]. И даже итерационный поиск был реализован еще в 1968 г. Герардом Сэлтоном в поисковой системе SMART (Salton's Magical Automatic Retriever of Text) [15]. Поэтому в процессе проведения реальных БИ была выработана технология без применения специальных программных средств поддержки и реализована в виде набора правил использования стандартных ПМ и средств операционных систем и офисных программ (технология НП). По всей видимости, технология НП не является единственно возможной, но следует полагать, что отличия возможны только в деталях.

6 Описание технологии, основанной на наборе правил

Работа исследователя в соответствии с технологией НП осуществляется поэтапно. Первый этап — собственно поиск (ПИП и КИП). По завершении

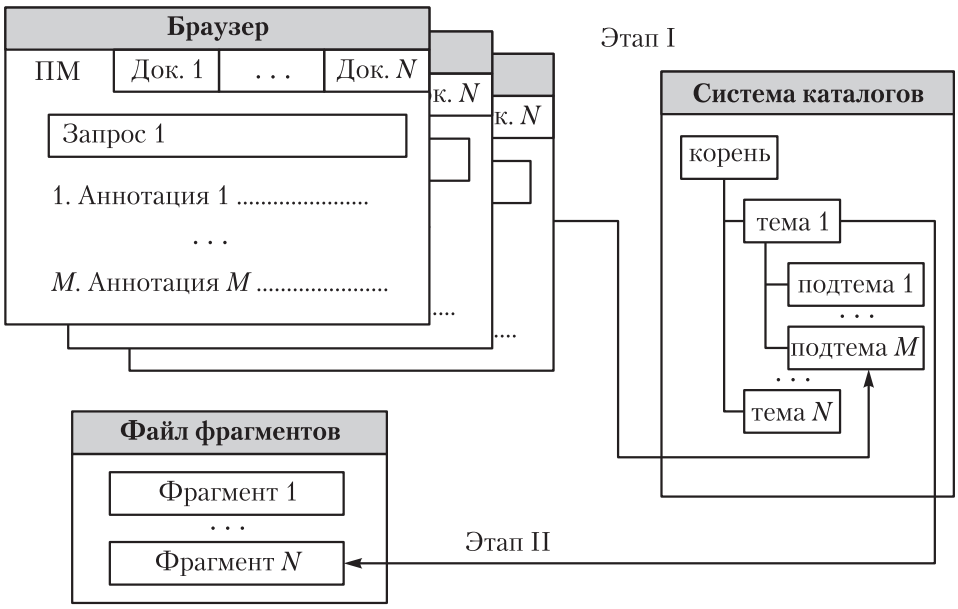


Рис. 1 Схема работы в соответствии с технологией НП

собственно поиска начинается следующий этап — оценка, сортировка и систематизация найденной информации. Завершение первого этапа происходит в одном из двух случаев:

(1) несомненное нахождение интересующей информации, полностью удовлетворяющей данную информационную потребность;

(2) отсутствие новых идей для создания новых запросов.

К сожалению, по опыту реальных БИ, первый вариант (несомненное нахождение интересующей информации) является более редким, чем второй. Но второй вариант (отсутствие новых идей) никоим образом не означает неудачу поиска, поскольку интересующая информация прямо или косвенно вполне может содержаться в найденных документах, но ее извлечение и осознание требует отдельной аналитической работы, для чего и служит второй этап поиска. Работа исследователя в соответствии с описываемой технологией схематически отражена на рис. 1.

6.1 Правила первого этапа

1. Поиск осуществляется итерационно в обычном браузере. В первой вкладке браузера открывается поисковая система (например, Яндекс). На каждой

итерации запрос переформулируется с учетом результатов предыдущих запросов.

2. Результат выдачи (список аннотаций документов) просматривается достаточно глубоко, не ограничиваясь первой страницей.
3. Документы, признанные по аннотации имеющими отношение к теме поиска, открываются в новой вкладке браузера пока без визуального просмотра.
4. При необходимости создать ветвление (когда соображения для новой формулировки запроса появляются до окончания просмотра списка аннотаций, соответствующего текущему запросу) открывается новое окно браузера, в первой вкладке которого открывается поисковая система и выполняется новый запрос.
5. По окончании просмотра списка аннотаций принимается решение о переформулировании запроса для следующей итерации. Для такого решения, как правило, достаточен анализ аннотаций. В редких случаях необходим просмотр самого документа.
6. При накоплении открытых документов во вкладках браузера сверх разумного количества происходит их предварительный просмотр и принятие решения о сохранении или удалении. Документы, признанные по результатам предварительного просмотра имеющими отношение к предмету исследования, сохраняются в некотором каталоге. Вкладки браузера с открытыми документами закрываются, и поиск продолжается.

6.2 Правила второго этапа

1. Каждый документ просматривается, при нахождении в документе фрагмента, важного с точки зрения исследования, этот фрагмент копируется в отдельный текстовый файл с указанием источника. Сам же документ переносится в библиотеку — систему иерархически организованных по темам каталогов на диске — либо удаляется, если при тщательном просмотре документ признается не содержащим интересных с точки зрения исследований или новых данных.
2. Для дальнейшей работы по теме исследований используется сформированный файл фрагментов, а также, при необходимости что-либо уточнить, библиотека документов.

7 Недостатки технологии, основанной на наборе правил

Описанная выше технология, несмотря на свою эффективность, содержит ряд существенных недостатков, приводящих к неоправданной трате временны́х ресурсов и повышающих вероятность пропуска важного документа:

1. В выдаче по запросу часто присутствуют одинаковые или одинаковые в существенной части аннотации. Это является следствием того, что в Интернете многие источники цитируют друг друга. Это существенно увеличивает несодержательный объем выдачи и тем самым затрудняет просмотр.
2. Найденные на предыдущих шагах итерации документы, относительно которых уже была сделана предварительная оценка их ценности и которые уже, возможно, присутствуют в открытых вкладках браузера, все равно с большой вероятностью попадут в выдачу на следующем шаге итерации, что также существенно «замусоривает» выдачу.
3. В выдачу попадают документы, не имеющие отношения к теме исследования только потому, что обычные ПМ в основном используют текстовый поиск, т. е. поиск по ключевым словам с учетом близости слов в тексте. Несмотря на то что в поисковые механизмы постоянно вводятся расширения, позволяющие учитывать семантику, до реализации полноценного семантического поиска еще очень далеко. В результате в выдаче часто встречаются аннотации, в которых ключевые слова оказываются несвязанными. Например, при поиске человека по ФИО (что типично для БИ) сравнительно часто выдаются предложения, где идет перечисление лиц, но запрашиваемые фамилия, имя, отчество относятся к разным лицам, которые расположены рядом [16]. Приводит это к тому же «замусориванию» выдачи.
4. При сохранении документа на диск, как правило, теряется его адрес в Интернете, который необходим, во-первых, для корректного оформления библиографических ссылок, а во-вторых, ресурс, содержащий документ, может быть интересен сам по себе и содержать иную важную для исследователя информацию.
5. Организация библиотеки сохраненных документов средствами файловой системы ОС влечет за собой следующие проблемы:
   - она часто требует ручного переименования файлов с целью присвоения им содержательных имен;
   - присвоенные имена могут быть слишком длинными для комфортного их просмотра и для программ резервного копирования;
   - сохраненный документ может представлять собой не один файл, а систему файлов и каталогов. Это затрудняет просмотр и вызывает проблемы при необходимости копирования или переименования документа;
   - система каталогов поддерживает только иерархическую организацию, которая входит в конфликт с мультитемностью документов.
6. Организация «выжимки» наиболее значимых фрагментов в виде текстового файла требует ручной привязки (в виде некоторой ссылки) фрагмента к документу, в котором он исходно содержался. Необходимость пересортировывать документы и менять их расположение на диске приводит к невозможности

использования ссылок, допускающих прямой переход по ним. Соответственно, поиск документа по ссылке должен производиться вручную.

7. Пересортировка фрагментов в текстовом файле затруднительна.

8 Требования к средствам поддержки технологии

Из вышесказанного вытекает настоятельная потребность в создании средств поддержки (СП) технологии БИП, позволяющих реализовать технологию поиска (технологию СП), свободную от перечисленных недостатков.

Средства поддержки должны обеспечивать:

- систематизацию задач пользователя, привязку поиска к задаче;
- поддержку итерационного ветвящегося поиска с просмотром списка аннотаций документов на каждой итерации;
- возможность просмотра документа по его аннотации;
- настраиваемую возможность показать на текущем и последующих шагах итерации в списке аннотаций из многих документов с совпадающими аннотациями только одного представителя;
- настраиваемую возможность для исследователя выделить в аннотации документа содержательную часть, с тем чтобы на текущем и последующих шагах итерации показать в списке аннотаций из многих документов, содержательные части аннотаций которых совпадают, только одного представителя;
- возможность задать список «стоп-слов» (включая выражения из нескольких слов), с тем чтобы аннотации, которые содержат хотя бы одно слово из этого списка, были исключены из отображения на текущем и последующих шагах итерации;
- возможность для исследователя отметить документ в выдаче как «потенциально интересный» (ПИ) или как «точно неинтересный» (ТН), с тем чтобы на последующих шагах итерации отмеченные документы можно было исключить из выдачи;
- автоматическое сохранение всех документов, помеченных как ПИ, в библиотеку с привязкой к задаче пользователя, к запросу и к их адресу в сети Интернет;
- полнотекстовый поиск по документам, сохраненным в библиотеке, их просмотр, маркировку тегами и поиск по тегам;
- выделение, сохранение, компоновку и просмотр фрагментов документов.

9 Описание технологии, опирающейся на средства поддержки

Работа исследователя в соответствии с новой технологией (технологией СП) схематически отражена на рис. 2. Действия пользователя во многом подобны действиям, предусмотренным технологией НП, но за счет того, что связи

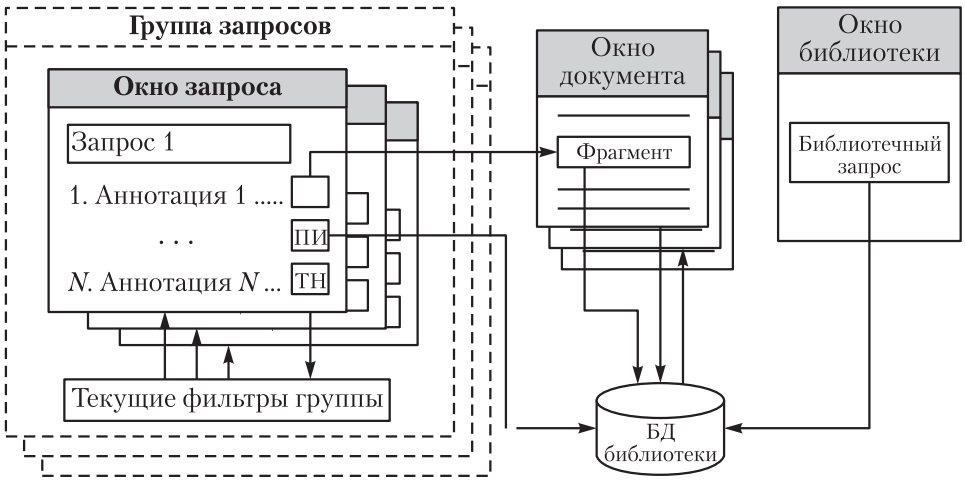


Рис. 2 Схема работы в соответствии с технологией СП

между окнами и связанными с ними объектами поддерживаются программно и пользователю не требуется удерживать их в памяти, явное разделение действий пользователя на два этапа уже не требуется. Все возможности технологии доступны пользователю в каждый момент времени.

Ниже описаны элементы программы и связанные с ними возможности, реализующие в совокупности новую технологию.

9.1 Вход в систему

При входе в систему пользователь выбирает исследование из списка или создает новое исследование. Это необходимо, поскольку несколько независимых исследований может вестись пользователем одновременно. При выборе существующего исследования восстанавливается состояние окон программы.

9.2 Работа с окном запросов

Окно служит для ввода интернет-запроса и отображения аннотаций результатов поиска с учетом текущих фильтров. Окна объединены в группы, соответствующие одной информационной потребности. Новый запрос может быть задан как в существующем окне, так и в новом окне запроса. Новое окно может быть открыто как в существующей группе, так и в новой группе. За счет этого обеспечивается итерационный ветвящийся поиск. Фильтры отображения едины для группы окон. Фильтрация (удаление аннотаций из просмотра) осуществляется по следующим критериям:

– совпадающие аннотации;
– аннотации с совпадающим ядром;
– аннотации, содержащие слова из списка стоп-слов;
– аннотации, уже помеченные как ТН;
– аннотации, уже помеченные как ПИ.

Документы, аннотации которых были помечены пользователем как ПИ, автоматически сохраняются в базе данных библиотеки. При этом для них сохраняются также их заголовок, расположение документа в сети Интернет (URL — uniform resource locator), тема текущего исследования, связанный поисковый запрос, текущая дата.

Для любой аннотации связанный документ может быть открыт в новом окне документов.

9.3 Окно документа

Окно служит для просмотра одного документа и его метаинформации:

– заголовка документа;
– связанного поискового запроса;
– URL;
– даты нахождения;
– комментария пользователя;
– списка тегов.

Заголовок, комментарий и список тегов могут быть отредактированы пользователем.

В документе также могут быть выделены и сохранены фрагменты текста, представляющие особый интерес для исследователя.

9.4 Окно библиотеки

Окно библиотеки позволяет осуществлять добавление документа в базу данных библиотеки из файла, отбор документов по различным поисковым критериям, их отображение в окне документов или их удаление.

Документы могут быть отобраны:

– по заголовку;
– по дате;
– по запросу;
– по URL;
– по тегам;
– полнотекстовым поиском в комментарии;
– полнотекстовым поиском во всем документе;
– полнотекстовым поиском в выделенных фрагментах документа.

10 Заключение

Предложенная технология позволяет существенно упростить и ускорить БИ в такой ее значимой части, как поиск информации в сети Интернет, т. е. сделать исследование более доступным для широкого круга не являющихся профессиональными историками и биографами пользователей, что очень актуально в связи со все увеличивающимся общественным интересом к частной, семейной истории. В настоящее время осуществляется прототипная реализация соответствующего инструментария. Средства поддержки поиска реализуются как надстройка над стандартными ПМ. Библиотека документов реализуется на базе системы управления базами данных.

Пробная эксплуатация технологии СП в реальных БИ должна выявить проблемы и перспективы ее дальнейшего развития.

Литература

1. *Божков О. Б., Боголюбов И. Н.* О неполноте исходной информации в генеалогиях // Социологический журнал, 2005. № 2. С. 68–77.
2. *Аленичев В. В., Аленичева Т. Н.* Методика поиска биографических сведений в историко-генеалогическом исследовании // Современные гуманитарные исследования, 2009. № 4. С. 34–37.
3. *Жуйков Д. А.* Основные методологические проблемы биографического исследования в современной историографии // Вестник Челябинского гос. ун-та, 2012. № 25(279). История. Вып. 52. С. 133–136.
4. *Гусева И. И.* Биография и автобиография как жанры исторического исследования // Наука и общество. — Саратов: СГСЭУ, 2013. С. 26–30.
5. *Вахромеева О. Б.* Биографика как вспомогательная историческая дисциплина // Актуальные проблемы гуманитарных и естественных наук, 2013. № 3. С. 81–83.
6. *Гринев Д. В.* Семантический поиск в Web // Системи обробки інформації (Системы обработки информации). — Харків: ХУПС, 2012. Вып. 8(106). С. 75–78. http://www.repository.hneu.edu.ua/jspui/bitstream/123456789/1480.
7. *Ашманов И., Иванов А.* Оптимизация и продвижение сайтов в поисковых системах. 3-е изд. — СПб.: Питер, 2011. 464 с.
8. *Зубова О. В.* Ищем корни свои. Управление Государственной архивной службы Самарской области, 2012. http://regsamarh.ru/external/media/files/info_dejatelnost/publikazii/genealogia.pdf.
9. *Козлова Л. А.* Биографическое исследование российской социологии: предварительные теоретико-методологические замечания // Социологический журнал, 2007. № 2. С. 59–87.
10. *Иконникова С. Н.* Биографика как часть исторической культурологии // Вестник СПбГУКИ, 2012. № 2(11). С. 6–10.
11. *Маркова Н. А.* Логика биографических фактов // Информатика и её применения, 2012. Т. 6. Вып. 2. С. 49–58.
12. *Адамович И. М., Бирюкова Т. К., Гершкович М. Ю., Долгополов В. С., Заикин М. Ю., Козлова Л. М., Ломовцева Г. Г., Пешков А. Н.* Средства повышения

эффективности поиска документов в локальных информационно упорядоченных подпространствах // Системы и средства информатики, 2006. Вып. 16. С. 289–320.

13. *Пальчунов Д. Е.* Решение задачи поиска информации на основе онтологий // Бизнес-информатика, 2008. № 1. С. 3–13.
14. *Адамович И. М., Волков О. И., Маркова Н. А.* Метод классификации информации на основе иерархических тегов и его реализация на примере семейного архивного фонда // Системы и средства информатики, 2012. Вып. 22. № 2. С. 146–156.
15. *Сэлтон Г.* Автоматизированная обработка, хранение и поиск информации / Пер. с англ. — М.: Советское радио, 1973. 580 с. (*Salton G.* Automatic information organization and retrieval. — New York, NY, USA: McGraw-Hill, 1968. 527 p.).
16. *Кузнецов И. П., Шарнин М. М., Мацкевич А. Г.* Интеллектуальные механизмы семантического поиска в сети Интернет // Системы и средства информатики, 2012. Т. 22. № 2. С. 129–145.

Поступила в редакцию 19.03.14

THE TECHNOLOGY OF INTERNET SEARCH AS A PART OF BIOGRAPHIC INVESTIGATION

I. M. Adamovich and O. I. Volkov

Institute of Informatics Problems, Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation

Abstract: The article examines the role and specificity of Internet search as a part of biographic investigation. The reasons of possible problems occuring during search are explored. With this purpose, the types of user's biographical information necessities were analyzed. The specific biographical information necessities were found and the type of search associated with them — "indirect search" was described. The requirements for Internet search technology which supports and simplifies indirect search are formulated. An example of such technology which does not use any special software and is based on a set of rules is described and analyzed. The requirements for special search support software were formulated as the result of analysis of faults of this technology. The technology of effective biographical search based on such software, free of analyzed faults and supporting storage and access to search results was described. This technology is meant for a wide range of users which are not professional historians or biographers. This is important today because of increasing public interest in family history.

Keywords: biographic investigation; technology of Internet search

DOI: 10.14357/08696527140212

References

1. Bozhkov O. B., and I. N. Bogoljubov. 2005. O nepolnote iskhodnoy informatsii v genealogiyakh [On incompleteness of initial information in genealogies]. *Sotsiologicheskiy Zhurnal* [Sociological Magazine] 2:68–77.
2. Alenichev, V. V., and T. N. Alenicheva. 2009. Metodika poiska biograficheskikh svedeniy v istoriko-genealogicheskom issledovanii [The methodology of biographical information search during the historical and genealogical investigation]. *Sovremennye Gumanitarnye Issledovaniya* [Modern Humane Research] 4:34–37.
3. Zhujkov, D. A. 2012. Osnovnye metodologicheskie problemy biograficheskogo issledovaniya v sovremennoy istoriografii [The main methodological problems of the biographical research in modern historiography]. *Vestnik Cheljabinskogo Gos. Un-ta* [Bulletin of Chelyabinsk State University]. 25:133–136.
4. Guseva, I. I. 2013. Biografiya i avtobiografiya kak zhanry istoricheskogo issledovaniya [Biography and autobiography as genres of historical studies]. *Nauka i Obshchestvo* [Science and Society] 1:26–30.
5. Vahromeeva, O. B. 2013. Biografika kak vspomogatel'naya istoricheskaya distsiplina [The biographical genre as an adjuvant historical branch of science]. *Aktual'nye Problemy Gumanitarnykh i Estestvennykh Nauk* [The Actual Problems of Humane and Natural Studies] 3:81–83.
6. Grinev, D. V. 2012. Semanticheskiy poisk v WEB [The semantic search in WEB]. *Sistemy Obrabotki Informatsii* [The Systems of Data Processing] 8(106). Available at: http://www.repository.hneu.edu.ua/jspui/handle/123456789/1480 (accessed March 27, 2014).
7. Ashmanov, I., and A. Ivanov. Optimizatsiya i prodvizhenie saytov v poiskovykh sistemakh [Web-sites optimization ans promotion for search engines]. SPb.: Piter Publs. 464 p.
8. Zubova, O. V. 2012. Ishchem korni svoi [Investigating of own antecedents]. Available at: http://regsamarh.ru/info_act/publishing/ (accessed March 27, 2014).
9. Kozlova, L. A. 2007. Biograficheskoe issledovanie rossiyskoy sotsiologii: Predvaritel'nye teoretiko-metodologicheskie zamechaniya [The biography research of Russian sociology: Preliminary theoretical and methodological notes]. *Sotsiologicheskiy Zhurnal* [Sociological Magazine] 2:59–87.
10. Ikonnikova S. N. 2012. Biografika kak chast' istoricheskoy kul'turologii [Biografical studies as a part of the historical cultural studies]. *Vestnik SPbGUKI* [Bulletin of Saint-Petersburg State University of Culture and Art]. 2(11):6–10.
11. Markova, N. A. 2012. Logika biograficheskikh faktov [A logic of biographical facts]. *Informatika i ee Primenenija — Inform. Appl.* 6(2):49–58.
12. Adamovich, I. M., T. K. Birjukova, M. Ju. Gershkovich, V. S. Dolgopolov, M. Ju. Zaikin, L. M. Kozlova, G. G. Lomovceva, and A. N. Peshkov. 2006. Sredstva povysheniya effektivnosti poiska dokumentov v lokal'nykh informatsionno uporyadochennykh podprostranstvakh [The means to improve the effectiveness of document searching in the local informationally ordered subspaces]. *Sistemy i Sredstva Informatiki — Systems and Means of Informatics* 16:289–320.
13. Pal'chunov, D. E. 2008. Reshenie zadachi poiska informatsii na osnove ontologiy [The solution based on the ontology of the information search problem]. *Biznes-Informatika* [Business Informatics] 1:3–13.

14. Adamovich, I. M., O. I. Volkov, and N. A. Markova. 2012. Metod klassifikatsii informatsii na osnove ierarkhicheskikh tegov i ego realizatsiya na primere semeynogo arhivnogo fonda [Method of information classification based on hierarchical tags and its implementation on the example of a family archive]. *Sistemy i Sredstva Informatiki — Systems and Means of Informatics* 22(2):146–156.
15. Salton, G. 1968. *Automatic information organization and retrieval*. New York, NY, USA: McGraw-Hill. 527 p.
16. Kuznecov, I. P., M. M. Sharnin, and A. G. Mackevich. 2012. Intellektual'nye mekhanizmy semanticheskogo poiska v seti Internet [Intellectual mechanisms for semantic searching in Internet]. *Sistemy i Sredstva Informatiki — Systems and Means of Informatics* 22(2):129–145.

Received March 19, 2014

Contributors

Adamovich Igor M. (b. 1934) — Candidate of Science (PhD) in trechnology, Head of Department, Institute of Informatics Problems, Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; Adam@amsd.com

Volkov Oleg I. (b. 1964) — leading programmer, Institute of Informatics Problems, Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; Volkov@amsd.com

ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ МОНИТОРИНГА НАЦИОНАЛЬНОЙ БЕЗОПАСНОСТИ В РЕГИОНАЛЬНОМ РАЗРЕЗЕ

Г. В. Лукьянов[1], *Д. А. Никишин*[2], *Г. Ф. Веревкин*[3]

Аннотация: Мониторинг и оценка национальной безопасности Российской Федерации (далее — НБРФ) предполагает обработку и учет нескольких десятков предусмотренных для этого разнообразных показателей. Априори предполагается, что показатели либо не подвержены региональной специфике, либо это влияние незначительно. Тем не менее, проведенный анализ показал, что значительная часть показателей выводится из региональных компонент и может быть получена только из региональных источников, причем эти показатели серьезно варьируют от региона к региону, а различия имеют не только количественный, но и качественный характер. Основными факторами региональной специфики выступают существенные различия в природных, климатических и геологических условиях, хотя некоторые эксперты отмечают и неравные «стартовые» возможности различных регионов. Еще одна особенность статистических показателей состоит в их достаточно формальном характере, иногда весьма опосредованно отражающем суть явления, которое они оценивают. И только углубленное изучение ситуации позволяет выяснить причину таких расхождений. Эти обстоятельства диктуют необходимость разработки специализированных методик, учитывающих региональную специфику при расчете основных (федеральных) показателей НБРФ. В качестве такого решения может быть использован региональный поправочный коэффициент (РПК), который позволял бы корректировать ряд наиболее значимых с точки зрения НБРФ статистических показателей в интересах достижения большей объективности.

Keywords: информационное обеспечение; мониторинг; региональная дифференциация; региональная специфика; национальная безопасность; административно-территориальное устройство; субъекты Федерации

DOI: 10.14357/08696527140213

1 Введение

Мониторинг и оценка НБРФ предполагает обработку и учет нескольких десятков предусмотренных для этого разнообразных показателей, отражающих

[1]Институт проблем информатики Российской академии наук, gena-mslu@mail.ru

[2]Институт проблем информатики Российской академии наук, dmnik@a170.ipi.ac.ru

[3]Институт проблем информатики Российской академии наук, gennadij.verevckin2012@yandex.ru

ситуацию в целом по стране. Многие из них формируются, в свою очередь, на основе агрегирования (в частности, усреднения, сложения и т. п.) аналогичных показателей, характеризующих положение дел в каждом из отдельно взятых субъектов Российской Федерации. Вместе с тем Россия — это самая большая по территории страна в мире. Она отличается от всех остальных стран уникальным сочетанием геополитических условий, для большинства из которых характерна высокая степень региональной дифференциации с точки зрения культурного и образовательного уровня, социально-экономического развития, политической активности населения, природно-климатических условий и некоторых других факторов [1].

Проведенный авторами статьи анализ показателей, рекомендованных к учету для оценки состояния НБРФ, позволяет провести их категоризацию по степени влияния на эти показатели региональной специфики. Действительно, ряд показателей либо не подвержен региональной специфике, либо это влияние столь незначительно, что учитывать его просто нецелесообразно. В качестве примеров можно привести такие показатели, как:

– уровень государственного внешнего и внутреннего долга;

– доля лекарственных средств импортного производства;

– уровень инфляции в стране.

Вместе с тем значительная часть показателей выводится исключительно из региональных компонент и может быть получена только из региональных источников. К таковым, например, относятся:

– уровень безработицы;

– доля населения с доходами ниже прожиточного минимума;

– относительная рождаемость.

Все эти показатели серьезно варьируют от региона к региону. Даже если Россию сравнить с другими странами, например с Китаем (31 провинция) или США (51 штат), все равно нельзя не признать, что они по совокупности геополитических факторов существенно уступают России с ее уникальной действительностью. Настоящая статья посвящена анализу степени влияния региональной специфики на НБРФ и предложениям по ее учету при разработке информационного обеспечения систем мониторинга НБРФ. Благоприятным фактором для учета региональной специфики при организации информационного мониторинга НБРФ является сложившаяся в России система статистических наблюдений, которая ряд показателей представляет в региональном разрезе. Это обстоятельство позволяет на достаточно большом временно́м интервале провести детальные исследования региональной специфики и ее влияния на НБРФ.

На основе результатов данного анализа в дальнейшем предполагается разработать концептуальные подходы к разработке информационного обеспечения систем мониторинга НБРФ с учетом региональной специфики.

2 Качественный «разрез» региональной специфики

В настоящее время различия российских регионов имеют не только ярко выраженный количественный, но и расширенный качественный характер. В частности, в Российской Федерации отмечается весьма контрастная региональная асимметрия по следующим направлениям:

1. «Север–Юг» (широтная зональность). В настоящее время так называемый Север, составляющий около половины территории России, дает много более половины ценного сырья, но освоен и заселен крайне неравномерно [2]. Юг России относительно невелик и небогат, но весьма плотно заселен.
2. «Запад–Восток». На европейскую Россию, занимающую лишь четверть территории страны, приходится до 80% ее населения и более 70% ВВП, в то время как восточные регионы демонстрируют низкую плотность населения и относительно небольшой уровень промышленного потенциала.
3. «Центр–периферия» («ядро–периферия»). Сегодня в Российской Федерации наблюдаются весьма значительные различия в уровне социально-экономического развития не только между столицей и остальными субъектами Федерации, но и между центрами этих субъектов и их территориями, причем различия внутри субъектов Российской Федерации зачастую сильнее, чем различия между субъектами. Это обстоятельство, наряду с прочими, приводит к оттоку населения с периферии и его перетеканию в крупные экономические, политические и культурные центры страны, особенно в Москву и Московскую область.

К этой картине качественного «разреза» региональной специфики следует добавить еще и тот факт, что трансформационный спад промышленности в регионах в результате политических и социально-экономических преобразований в разных регионах также оказался весьма дифференцированным. В результате этого спада соотношение между Севером и Югом по показателю среднедушевых доходов населения достигло пятнадцатикратного разрыва и в 2000 г. лежало в диапазоне 270–4017 руб. в месяц на одного жителя.

В последующем ситуация по этому показателю существенно улучшилась, тем не менее значительные диспропорции между регионами все же сохранились до сих пор. Все эти диспропорции привели к необратимым явлениям в межкультурных, межэтнических и межрелигиозных отношениях, имеющих ярко выраженную региональную специфику, которую необходимо отражать при информационном мониторинге НБРФ.

3 Оценка степени региональной дифференциации

Одной из наиболее значимых с точки зрения НБРФ политических особенностей России является ее довольно сложное, не имеющее мировых аналогов

Таблица 1 Особенности категорий субъектов

Субъект	Особенности
Республика	– Охарактеризована как «государство» – Вправе устанавливать свои государственные языки
Край, область, город федерального значения	– Статус определяется Конституцией России и собственным уставом*
Автономная область	– Статус определяется Конституцией России и собственным уставом* – Может быть принят федеральный закон об автономной области
Автономный округ	– Статус определяется Конституцией России и собственным уставом* – Может быть принят федеральный закон об автономном округе – Отношения автономных округов, входящих в состав края или области, могут регулироваться федеральным законом и договором между соответствующим автономным округом и краем или областью

*Устав принимается законодательным представительным органом соответствующего субъекта.

федеральное устройство, в соответствии с которым Российская Федерация в своем составе объединяет 83 субъекта[1]. Причем входящие в состав Российской Федерации субъекты крайне неоднородны как по своему административному устройству, так и по основным параметрам оценки (территории, населенности и т. п.), что, собственно говоря, и является главным отличительным признаком. Так, на начало 2014 г. в состав Российской Федерации входили 21 республика, 9 краев, 46 областей, два города федерального значения, одна автономная область и четыре автономных округа.

Согласно действующей Конституции Российской Федерации, все входящие в ее состав субъекты равноправны во взаимоотношениях с федеральными органами власти (без права выхода из состава Российской Федерации), но для каждой категории предусмотрены некоторые правовые особенности, которые отражены в табл. 1.

Таким образом, Конституция Российской Федерации, с одной стороны, декларирует равенство всех субъектов Российской Федерации, а с другой — закрепляет их разный статус, что содействует асимметричному региональному развитию. Такое формальное деление субъектов на категории (не говоря уже о других факторах) ведет к присвоению некоторыми регионами полномочий федеральной власти и предполагает потенциально разную степень влияния отдельных

[1]По состоянию на 1 января 2014 г.

Таблица 2 Соотношение между городским и сельским населением

Постоянное население на 01.01.2010	Все население, тыс. чел.	В том числе, тыс. чел. (в % к населению)	
		Город	Село
Российская Федерация	141 914,5	103 705,3 (73,1)	38 209,2 (26,9)
Центральный федеральный округ	37 118,0	30 032,0 (80,9)	7 086,0 (19,1)
Северо-Западный федеральный округ	13 437,1	11 085,5 (82,5)	2 351,6 (17,5)
Южный федеральный округ	13 713,5	8 516,2 (62,1)	5 197,3 (37,9)
Северо-Кавказский федеральный округ	9 254,9	4 525,9 (48,9)	4 729,0 (51,1)
Приволжский федеральный округ	30 109,4	21 164,3 (70,3)	8 945,1 (29,7)
Уральский федеральный округ	12 280,1	9 736,5 (79,3)	2 543,6 (20,7)
Сибирский федеральный округ	19 561,1	13 861,6 (70,9)	5,5 (29,1)
Дальневосточный федеральный округ	6 440,4	4 783,3 (74,3)	1,1 (25,7)

субъектов Российской Федерации на НБРФ, что отражается на нескольких значимых с точки зрения НБРФ показателях. На эту дифференциацию по категориям входящих в состав страны субъектов накладываются крайне разнообразные региональные условия, связанные со специфическим геополитическим положением каждого из субъектов России.

Наглядно высокую степень региональной дифференциации демонстрирует соотношение между городским и сельским населением, как представлено в табл. 2 [3].

Из приведенных данных, в частности, следует, что доля городского населения в Северо-Западном федеральном округе почти в два раза выше, чем в Северо-Кавказском федеральном округе и это не может не определять существенную разницу, например, в политической активности жителей обоих округов. В свою очередь, это обстоятельство определяет разную степень влияния на НБРФ обоих указанных федеральных округов. Если же этот показатель рассмотреть применительно к некоторым административно-территориальным единицам с особым статусом, как представлено в табл. 3 [3], то ситуация окажется еще более впечатляющей.

Как видно из табл. 3, в Коми-Пермяцком округе доля городского населения составляет менее 25% со всеми вытекающими из этого последствиями. В то же время Таймырский автономный округ по этому показателю приближается к среднему значению по Российской Федерации.

Наиболее значимым фактором (после природно-географических), обострившим региональную дифференциацию и даже межрегиональные противоречия, стал переход России на рельсы рыночной экономики. В России возникла беспрецедентная неравномерность регионального развития, представляющая собой прямую угрозу ее территориальной целостности, вплоть до возникновения локальных вооруженных конфликтов между центром и региональными сепаратистскими группировками.

Таблица 3 Административно-территориальные единицы с особым статусом

Постоянное население на 01.01.2010	Все население, тыс. чел.	В том числе, тыс. чел. (в % к населению)	
		Город	Село
Коми-Пермяцкий округ	127,1	30,7 (24,2)	96,4 (75,8)
Агинский Бурятский округ	78,3	31,3 (39,9)	47,0 (60,1)
Таймырский автономный округ	36,6	24,5 (66,9)	12,1 (33,1)
Эвенкийский автономный округ	16,3	5,4 (33,0)	10,9 (67,0)
Усть-Ордынский Бурятский округ	135,3	— (—)	135,3 (100,0)
Корякский округ	20,4	5,6 (27,5)	14,8 (72,5)

Таким образом, в некоторых случаях региональная специфика является ярко выраженной, и это обстоятельство наталкивает на мысль о разной степени влияния на НБРФ некоторых показателей в различных регионах Российской Федерации. Это обстоятельство, в свою очередь, предполагает разработку специализированных методик расчета ряда показателей НБРФ и учета региональной специфики при разработке информационного обеспечения систем мониторинга НБРФ.

4 Формальный характер статистических показателей

Еще одна особенность статистических показателей состоит в их достаточно формальном характере, весьма опосредованно отражающем суть самого явления, которое они оценивают, т. е. показатели сами по себе не всегда отвечают на вопрос о причинах их низких или высоких значений. К примеру, при анализе расходов на прокладку коммуникаций бросается в глаза тот факт, что стоимость единицы строительства от региона к региону может отличаться почти в десять раз. И только углубленное изучение ситуации позволяет выяснить причину таких расхождений. В данном случае основным фактором выступают серьезные отличия в природных, климатических и геологических условиях, хотя некоторые эксперты отмечают и неравные «стартовые» возможности различных регионов.

Еще один важнейший с точки зрения НБРФ показатель — это уровень жизни населения, на который существенное влияние оказывают климатические условия. Действительно, даже при равной заработной плате и равных ценах на товары люди, живущие в условиях холодного климата, вынуждены нести значительно бóльшие расходы на приобретение теплой одежды, оплату услуг по обогреву и содержанию жилья, чем жители регионов с более мягким климатом. Еще хуже ситуация в регионах с резко континентальным климатом, где требуется постоянно иметь несколько комплектов одежды и нести значительные расходы на обогрев и кондиционирование помещений (жилых или офисных). Или, например, гидрология — при одинаковой доступности водных источников вода в одном случае может быть пригодна для питья, а в другом — совершенно неприемлемой

для организма человека, что вынуждает применять дорогостоящую технологию очистки воды.

В общемировой практике оценки благосостояния населения важным показателем является наличие (или отсутствие) в семье автомобиля. В отличие от бытовой техники (холодильник, телевизор, микроволновая печь, пароварка), которая распространена повсеместно, покупка и содержание автомобиля остается уделом достаточно обеспеченных граждан. Однако невозможно только со статистических позиций объяснить географическую картину распределения автомобилей в расчете на одну семью (или на душу населения). Среди лидеров по количеству автомобилей на душу населения выступают такие регионы, как Приморский край, Сахалинская и Калининградская области, а среди «отстающих» — северные автономные округа. Если исходить только из статистических показателей, может показаться, что благосостояние жителей, например, Сахалинской области явно выше, чем обитателей Севера. Однако это не так, здесь начинают играть роль совсем другие факторы, причем весьма далекие от экономической специфики, например культура, история или даже религия. Так, жители северных районов нашей страны в автомобилях просто не нуждаются: для них большее значение имеют сани, нарты, собачьи упряжки.

Приведенные примеры убедительно диктуют необходимость введения некоторого РПК, который позволял бы корректировать ряд наиболее значимых с точки зрения НБРФ статистических показателей в интересах достижения большей объективности. При разработке конкретных значений РПК уместно опираться на результаты исследований Института географии РАН по изучению условий Крайнего Севера и приравненных к ним местностей, охватывающих 60% территории Российской Федерации. В ходе этих исследований было выделено пять зон условий, правда, только по климатическим факторам [4].

Например, ученые из Института географии РАН в качестве индикатора, характеризующего уровень жизни в регионе, предлагают транспортную доступность, а именно: соотношение между протяженностью автомобильных дорог и средним расстоянием, на которое ежедневно перемещается население региона. С этим можно отчасти согласиться, исходя из известной русской пословицы и понимания того, что транспортная освоенность территории в определенной степени служит мерилом экономической деятельности. Действительно, состояние дорожной сети и транспортной системы в целом является решающим для доступа к таким благам цивилизации, как медицинские, образовательные, культурные и административные учреждения (нотариальные конторы, театры, бассейны, школы, вузы, библиотеки). В России 75% малых городов удалены от областного центра на 80 и более километров, а 10% — на расстояние более 500 км. Вместе с тем такой подход возможен только для проверки реализуемости общей концепции, однако в последующем требует более глубокого изучения этой проблемы.

Некоторые из отмеченных различий имеют глубокие исторические корни, обусловленные тем, что в прошлые века в Россию вошли территории и социальные общности, находящиеся на разных стадиях исторического и социально-эко-

номического развития [5]. В советское время острота отдельных различий сглаживалась путем плановой политики выравнивания уровней социально-экономического развития как граждан, так и регионов. Таким образом, учет только природно-климатических условий может быть лишь первым шагом в общей методологии оценки региональной специфики в интересах разработки информационного обеспечения систем мониторинга НБРФ.

5 Показатели национальной безопасности Российской Федерации в региональном аспекте

Исходя из международного опыта, в частности опыта деятельности Департамента национальной (внутренней) безопасности США (Department of Homeland Security), наиболее зависимыми от региональной специфики (по преобладающему числу показателей) сферами оценивания национальной безопасности являются:

- экономическая сфера (за исключением финансовой системы);
- социальная сфера;
- духовная и информационная сфера[1];
- экологическая сфера;
- сфера общественной безопасности.

Зависимыми по отдельным показателям являются:

- внутриполитическая сфера;
- сфера образования и науки.

Практически независимыми от региональной специфики являются:

- международная сфера;
- военная и оборонно-промышленная сфера;
- финансовая система экономической сферы.

Из анализа результатов научных изысканий и действующих правовых норм [6] следует, что обязательными показателями для мониторинга и оценки НБРФ, которые наиболее существенно влияют на состояние НБРФ и которые зависят от региональной специфики, наряду с прочими могут быть следующие:

- доля населения, включенного в акции протеста с требованиями политического характера;
- уровень безработицы;
- доля населения с доходами ниже прожиточного минимума;
- соотношение доходов 10% наиболее и 10% наименее обеспеченного населения (децильный коэффициент);

[1] Следует отметить, что «духовная и информационная сфера» в настоящее время представлена, в основном, информационными показателями.

Таблица 4 Доля населения с доходами ниже прожиточного минимума

Регион	Год			
	2007	2013	2017	2020
Мурманская область	20,2%	18,0%	15,0%	11,0%
Архангельская область (без Ненецкого автономного округа)	23,2%	18,1%	15,4%	10,3%
Ненецкий автономный округ	5,7%	4,6%	3,7%	2,3%
Республики Саха (Якутия)	20,1%	17,3%	14,9%	11,2%
Чукотский автономный округ	12,4%	12,1%	11,7%	11,5%
Арктическая зона России	14,9%	12,5%	10,6%	8,0%

Таблица 5 Уровень безработицы

Регион	Год			
	2007	2013	2017	2020
Мурманская область	5,6%	7,8%	5,5%	4,2%
Архангельская область	7,3%	6,5%	5,3%	4,1%
Ненецкий автономный округ	6,4%	5,7%	4,2%	2,5%
Республика Саха (Якутия)	7,4%	4,3%	3,6%	3,1%
Чукотский автономный округ	3,6%	3,3%	2,9%	2,5%
Арктическая зона Российской Федерации	5,6%	5,1%	3,9%	3,0%

– состояние окружающей среды и экологии в целом;

– доля населения, проживающего на территориях, на которых состояние окружающей среды не соответствует нормативам качества;

– уровень преступности.

Региональную специфику демонстрирует показатель доли населения с доходами ниже прожиточного минимума от общей численности населения в некоторых периферийных регионах, как это представлено в табл. 4 [7].

Из приведенных данных следует, что по плану на 2017 г. все периферийные регионы должны иметь примерно одинаковые значения доли населения с доходами ниже прожиточного уровня. Тем не менее резким контрастом выступает Ненецкий автономный округ, хотя он формально мало чем отличается от Архангельской области.

Не менее впечатляющую картину представляет и распределение безработицы по регионам, как это видно из табл. 5 [3].

Как следует из данных, приведенных в табл. 5, самый высокий уровень безработицы в 2017 г. предполагается в Мурманской области, где он почти в 2 раза превышает этот показатель для Чукотского автономного округа. К 2020 г. планируется существенное выравнивание этого показателя, поэтому при последующей

переработке количества и качества индикаторов возможно перераспределение сфер по степени зависимости от региональной специфики.

6 Выводы

Приведенный материал убедительно демонстрирует значение учета региональной специфики как минимум по двум существенным с точки зрения обобщенной оценки НБРФ направлениям:

(1) диспропорции в региональных показателях настолько велики, что вычисление обобщенного показателя традиционным способом усреднения приводит к последующей искаженной оценке НБРФ;
(2) некоторые региональные показатели могут оказать на НБРФ в целом значительно большее влияние, чем традиционные показатели федерального уровня, например доля боеготовых соединений.

Таким образом, при разработке концептуальных подходов к созданию информационного обеспечения систем мониторинга НБРФ в большинстве случаев необходимо использовать не показатели, обобщенные органами государственной статистики и другими федеральными органами исполнительной власти, а исходные нередуцированные информационные ресурсы с региональной, временно́й и иной детализацией [8]. Так, из официальных информационных источников следует, что к 2020 г. предполагается сглаживание региональной дифференциации по крайней мере по наиболее значимым показателям, таким как доля населения с доходами ниже прожиточного минимума и уровень безработицы. Однако если отойти от статистических показателей и вникнуть в суть проблемы, то выяснится, что такое «выравнивание» достигается не за счет роста экономической активности отстающих регионов, а путем расширения масштабов дотационной деятельности. Скорее всего, такой подход приведет к еще бо́льшим социально-экономическим и политическим региональным перекосам, углублению межрегиональных противоречий, а значит, применять обобщенные статистические показатели для оценки НБРФ в данном случае просто неприемлемо. В связи с этим, возможно, в некоторых случаях возникнет необходимость сбора дополнительных статистических данных с требуемой детализацией.

Литература

1. Формирование и развитие системы регионального управления в России. http://www.economy-web.org/?p=451.
2. *Поздняков А., Лавровский Б., Масаков В.* Политика регионального выравнивания в России (основные подходы и принципы) // Вопросы экономики, 2000. № 10. С. 74–92.
3. Социально-демографический портрет России: По итогам Всероссийской переписи населения 2010 года. — М.: НИЦ Статистика, 2012. 184 с.
4. *Ласкин А.* Изучение региональных различий уровня жизни населения Российской Федерации. http://www.irex.ru/press/pub/polemika/13/las.

5. *Николаев И. А., Точилкина О. С.* Экономическая дифференциация регионов: оценки, динамика, сравнения. Аналитический доклад. — М.: Accountants & busines advisers, 2011. 28 с.
6. Стратегия национальной безопасности Российской Федерации до 2020 года, утвержденная Указом Президента Российской Федерации № 537 от 12.05.2009.
7. Показатели развития автономных округов до 2020 года. http://do.gendocs.ru/docs/index-67577.html?page=16, http://do.gendocs.ru/docs/index-67577.html?page=17.
8. Проблемы национальной безопасности: экспертные заключения, аналитические материалы, предложения / Под общ. ред. Н. П. Лаверова. — М.: Наука, 2008. 459 с.

Поступила в редакцию 20.11.13

COMMUNICATORY PROVISIONS FOR MONITORING OF NATIONAL SECURITY FROM THE REGIONAL POINT OF VIEW

G. V. Lukyanov, D. A. Nikishin, and G. F. Verevkin

Institute of Informatics Problems, Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation

Abstract: Monitoring and estimating of national security of the Russian Federation (NSRF) assumes processing of several tens of various parameters envisioned for this purpose. *A priori* it is supposed that these parameters are not subject to regional specificity or this influence is insignificant. The analysis carried out by the authors has shown that a significant part of parameters is injected from the regional components and can be received only from regional sources. Thus, these parameters vary seriously from region to region and their distinctions have not only quantitative, but also qualitative character. Essential distinctions in natural, climatic, and geological conditions are the major factors of regional specificity though some experts also specify unequal "starting" capabilities of various regions. One more feature of general statistics is reflected in its quite formal character, sometimes rather indirectly reflecting the essence of the phenomenon which the appropriate indices estimate. Only a profound study of the situation allows to straighten out the reason of such divergences. These circumstances dictate the necessity of working out specialized techniques considering regional specificity while calculating basic (federal) parameters of NSRF. One of the solutions could be applying the regional correction coefficient which would allow to correct a number of statistics that are the most significant from the point of view of NSRF in order to achieve better objectivity.

Keywords: communicatory provisions; monitoring; regional differentiation; regional features; national security; administrative and territorial structure; federation subject

DOI: 10.14357/08696527140213

References

1. Formirovanie i razvitie sistemy reginal'nogo upravleniya v Rossii [Shaping and development of regional management in Russia]. Available at: http://www.economyweb.org/?p=451 (accessed March 17, 2014).
2. Pozdniakov, A., B. Lavrovski, and V. Masakov. 2000. *Politika regional'nogo vyravnivaniya v Rossii (osnovnye podkhody i printsipy)* [Policy of regional leveling in Russia (main approaches and principles)]. Moscow: Voprosy Economiki 10:74–92.
3. 2012. *Sotsialno-demographicheskiy portret Rossii. Po itogam Vserossiyskoy perepisy naseleniya 2010* [Social and demographic portrait of Russia. Results of all-Russian census in 2010]. Moscow: NITZ Statistika. 184 p.
4. Laskin, A. Izuchenie regional'nykh razlichiy urovnya zhizni naseleniya Rossiskoy Federatsii [Regional differences study of living standards in the Russian Federation]. Available at: http://irex.ru/press/polemica/13/las/ (accessed March 17, 2014).
5. Nikolaev, I. A., and O. S. Tochilkina. 2011. Economicheskaya differentsiatsiya regionov: ozenki, dinamika, sravneniya. Analiticheskiy doklad [Economical differentiation of regions: Estimations, dynamics, comparisons. Analitical report]. Moscow: Accountants & business advisers. 28 p.
6. Strategiya natsional'noy bezopasnosti Rossii do 2020. Utverzhdena ukazom Prezidenta Rossiyskoy Federatsii No. 537, 12.05.2009 [Strategy of national security of Russia till 2020. Adopted by decree of the President of the Russian Federation No. 537, 12.05.2009].
7. Pokazateli razvitiya avtonomnykh okrugov do 2020 [Indices of autonomous districts development till 2020]. Available at: http://do.gendocs.ru/docs/index-67577.html?page=16; http://do.gendocs.ru/docs/index-67577.html?page=17 (accessed March 17, 2014).
8. Laverov, N. P., ed. 2008. *Problemy natsionalnoy bezopasnosti: Ekspertnye zaklyucheniya, analiticheskie materially, predlozheniya* [*National security problems: Peer-view conclusions, analytical materials, proposals*. Moscow: Nauka. 459 p.

Received November 20, 2013

Contributors

Lukyanov Gennady V. (b. 1952) — Candidate of Military Science (PhD), associate professor; Head of Laboratory, Institute of Informatics Problems, Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; gena-mslu@mail.ru

Nikishin Dmitry A. (b. 1976) — Candidate of Science (PhD) in technology, Head of Laboratory, Institute of Informatics Problems, Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; dmnik@a170.ipi.ac.ru

Verevkin Gennady F. (b. 1934) — Candidate of Science (PhD) in technology, senior scientist, Institute of Informatics Problems, Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; gennadij.verevckin2012@yandex.ru

МЕТОДЫ ИНТЕГРАЦИИ ОБЛАЧНЫХ СЕРВИСОВ НА ПРИМЕРЕ ЗДРАВООХРАНЕНИЯ

Г. Я. Илюшин[1], *В. И. Лиманский*[2]

Аннотация: Проведен анализ традиционных подходов к интеграции множества независимых приложений в единую информационную систему (ИС), рассмотрены их достоинства и недостатки. Изложены подходы к построению эффективного решения по интеграции приложений в рамках единой ИС и сформулированы требования для решения поставленной задачи на базе построения специального интеграционного слоя ИС. Указанное решение построено на принципах сервисно-ориентированной архитектуры, позволяет использовать современные интеграционные продукты различных компаний и предлагает дополнительные архитектурные средства масштабирования, обеспечивающие возможность оптимальной загрузки программно-аппаратных средств интеграционного слоя, начиная от небольших ИС уровня предприятия и вплоть до масштабных, территориально распределенных ИС. В частности, показана возможность и целесообразность применения рассмотренного интеграционного слоя при реализации единой государственной ИС в области здравоохранения (ЕГИСЗ) для решения задачи интеграции ее компонентов в единую ИС.

Ключевые слова: информационные системы; сервисно-ориентированная архитектура; SOAP; веб-сервисы; интеграция компонентов ИС; промежуточный слой; единая государственная информационная система в сфере здравоохранения; ЕГИСЗ

DOI: 10.14357/08696527140214

1 Введение

Попытки решения задач организации электронного взаимодействия между разнородными ИС как внутри ведомств, так и между ведомствами за последние годы предпринимались неоднократно. Однако ни одна из этих попыток не увенчалась полным успехом. Одной из причин неудач является попытка разработчиков реализовать взаимодействие унаследованных систем с ядром системы интеграции «в лоб» — путем разработки множества независимых облачных сервисов, решающих отдельные подзадачи взаимодействия. Пользователям этих сервисов (модернизируемым разнородным ИС) предлагается использовать новые сервисы

[1]Институт проблем информатики Российской академии наук, ilushin@atik.ru
[2]Институт проблем информатики Российской академии наук, vlimansky@ipiran.ru

и технологии информационной безопасности взамен ранее разработанных и уже функционирующих API (application programming interface). Таким образом, основные затраты на интеграцию неявно переносятся на модернизацию действующих систем и комплексную отладку нового интегрированного программного продукта. Размер этих трудозатрат существенным образом зависит от используемых базовых программных продуктов и предлагаемых форматов сервисов.

При написании статьи авторы использовали собственный опыт, полученный в процессе реализации в ИПИ РАН нескольких систем интеграции множества разнородных приложений, среди которых:

- подсистема интеграции информационных ресурсов ОВД ФТП ЕИП (2007–2009 гг.) [1–3];
- система бронирования медицинских услуг в лечебных учреждениях г. Москвы «Удаленная регистратура» (2003–2005 гг.) [4] в рамках проекта «Электронная Москва» (функциональный аналог современной подсистемы ФЭР ЕГИСЗ);
- более ранние проекты, в том числе система бронирования медикаментов [5].

Авторы также использовали результаты экспертизы реализации первой очереди системы межведомственного электронного взаимодействия (СМЭВ) и общедоступную информацию о реализации ЕГИСЗ.

2 Способы организации взаимодействия компонентов информационной системы

В сервисно-ориентированной архитектуре существует два метода организации взаимодействия интегрируемых компонентов системы. Первый заключается в организации прямого взаимодействия компонентов по принципу «точка–точка». Второй — во взаимодействии всех компонентов через «единый узел». Как правило, на начальных этапах автоматизации выбирается первый путь как наиболее простой в реализации и не требующий разработки дополнительного программного обеспечения. Суть данного метода заключается в следующем: отдельные компоненты ИС предоставляют сервисы, являющиеся интерфейсами к их бизнес-логике. Взаимодействие между компонентами реализовано в рамках бизнес-процессов, на отдельных шагах которых осуществляются вызовы того или иного сервиса: от сервиса-потребителя одного компонента к сервису-поставщику другого компонента. Когда взаимодействующих компонентов становится много, то контроль и управление функционированием компонентов усложняется, а стоимость сопровождения интегрированной таким образом ИС становится неприемлемо высокой. Следует также учитывать, что в работе ИС могут использоваться не только собственные сервисы-поставщики (сервисы компонентов своей ИС), но и сервисы-поставщики сторонних (внешних) ИС, которые в общем случае «играют» по своим правилам.

Организация взаимодействия программных компонентов путем прямого обращения сервиса-потребителя одного компонента к сервису-поставщику другого не желательна для целостной системы, поскольку имеет ряд существенных изъянов и недостатков:

– невозможно создание централизованной системы разграничения доступа к сервисам-поставщикам компонентов своей ИС, а также к сервисам-поставщикам сторонних ИС. В данном случае все средства разграничения доступа будут сосредоточены в сервисах-поставщиках компонентов;

– отсутствует возможность организации централизованного контроля состояния (мониторинга) функционирования и оперативного управления доступностью сервисов-поставщиков;

– невозможна организация системы централизованного протоколирования информационного взаимодействия компонентов, включая обработку возможных ошибок взаимодействия, с целью последующего получения статистических данных и проведения аудита.

Второй метод интеграции компонентов ИС заключается в организации взаимодействия всех компонентов системы через единый узел. Для построения систем интеграции компонентов ИС по этому методу рынок программного обеспечения предлагает достаточно широкий выбор продуктов. В частности, авторам несколько лет назад довелось принять непосредственное участие в реализации похожего проекта на платформе Microsoft [6]. С тех пор программные платформы интеграции существенно усовершенствованы. Рассмотрим, как можно решить задачу интеграции компонентов ИС через единый узел с использованием современных продуктов компании Oracle в сервисно-ориентированной архитектуре (SOAP).

Продукт компании Oracle Web Services Manager (WSM) — это комплексное средство управления решениями в SOAP-архитектуре. Оно позволяет централизованно задавать политики управления работой веб-сервисов (политики доступа, политики протоколирования, политики распределения нагрузки и др.), а затем применять их к веб-сервисам без внесения каких-либо изменений в сами сервисы [7–9]. Кроме того, Oracle WSM собирает данные мониторинга для оценки доступности и качества обслуживания. Применение Oracle WSM повышает управляемость и качество мониторинга веб-сервисов. Ключевыми компонентами Oracle WSM являются:

– менеджер политики;

– компоненты применения политики (агенты и шлюзы).

Для обеспечения максимальной гибкости Oracle WSM предоставляет два вида компонентов применения политики: шлюзы политик (Policy Gateways) и агенты политик (Policy Agents). Шлюзы политик (рис. 1) перехватывают запросы к сервисам с целью применения политик и устанавливаются перед группой приложений или сервисов:

Рис. 1 Защищенный доступ к веб-сервису с использованием шлюза

Шаг 1. Клиент отсылает запрос веб-сервису.

Шаг 2. Шлюз перехватывает запрос, применяет политики доступа (например, дешифрование, верификация, аутентификация, авторизация) и отправляет запрос веб-сервису.

Шаг 3. Веб-сервис обрабатывает запрос и возвращает ответ.

Шаг 4. Шлюз перехватывает ответ, применяет политики доступа (например, шифрование) и передает ответ клиенту.

Агенты политик (рис. 2) обеспечивают дополнительный дифференцированный уровень безопасности и размещаются на веб-сервисах:

Шаг 1. Клиент отсылает запрос веб-сервису.

Шаг 2. Агент перехватывает запрос, применяет политики доступа (например, дешифрование, верификация, аутентификация, авторизация) и отправляет запрос веб-сервису.

Шаг 3. Веб-сервис обрабатывает запрос и возвращает ответ.

Шаг 4. Агент перехватывает ответ, применяет политики доступа (например, шифрование) и передает ответ клиенту.

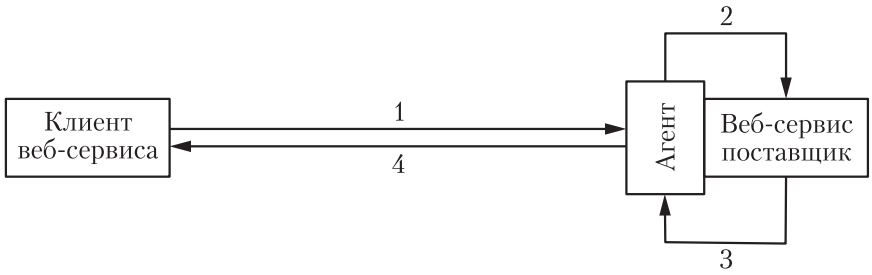


Рис. 2 Защищенный доступ к веб-сервису с использованием агента сервиса

Еще один продукт интеграционной платформы Oracle, Service Registry, предназначен для создания репозитория сервисов. Основные функции репозитория:

– публикация сервисов и связанных с ними метаданных;

– поиск сервисов по большому числу поисковых критериев;

– классификация сервисов на основе произвольных критериев;

– уведомление об изменениях, произошедших с сервисом на любом из этапов его жизненного цикла.

Рассмотренные продукты компании Oracle, использованные, в частности, при создании в рамках реализации концепции электронного правительства [10] системы межведомственного электронного взаимодействия [11], позволяют реализовывать интеграцию компонентов ИС через «единый узел», осуществлять централизованное управление сервисами интегрируемых компонентов ИС, обеспечивать назначение и применение нужных политик доступа, осуществлять мониторинг, сбор статистики и проведение аудита. Вместе с тем такому решению присущ целый ряд недостатков:

1. Один компонент ИС взаимодействует, как правило, с несколькими сервисами-поставщиками, для каждого из которых в составе данного компонента необходимо реализовать свой специфический клиент доступа к каждому сервису-поставщику.
2. Различные компоненты ИС могут взаимодействовать с одним и тем же сервисом-поставщиком, поэтому в составе различных компонентов должны быть реализованы клиенты доступа к этим сервисам-поставщикам, в результате чего одни и те же решения «размазываются» по компонентам ИС.
3. При использовании для разработки клиента сервиса-поставщика инструментальных средств, отличных от средств разработки самого сервиса, или при размещении сервиса и его клиента на разных программных платформах процесс создания клиента далеко не всегда сводится к простой компиляции его WSDL-схемы. Часто после генерации клиента по WSDL-схеме требуется «ручная доводка», т. е. по результатам тестирования должна осуществляться корректировка кода клиента и/или изменение настроек его конфигурационных файлов. Например, при обращении к веб-сервису Windows .NET со стороны Java-клиента:
   – несоответствия в требованиях к составу заголовков HTTP-пакета приводят к отказу от приема сообщения веб-сервером IIS (Internet Information Service);
   – сериализация массивов и списков в параметрах методов в теле;
   – Soap-сообщения могут иметь разные схемы и др.
4. При изменении сервиса-поставщика или при переходе к его новой версии может потребоваться корректировка всех компонентов ИС, в которых этот сервис-поставщик был использован.
5. При обращении к сервисам-поставщикам внешних ИС требуется создавать дополнительный шлюз, который сохранял бы управление над сервисом (политики безопасности, мониторинг, протоколирование) в исходной ИС, но политики доступа которого учитывали бы специфические требования внешнего сервиса-поставщика.

3 Интеграционный слой как база интеграции компонентов информационных систем

Разработаны технические решения по интеграции компонентов ИС, основанные на внедрении в систему специального интеграционного (промежуточного) слоя, реализующего функции контроля и управления всеми взаимодействиями между компонентами ИС и позволяющего устранить или существенно минимизировать перечисленные выше недостатки интеграции компонентов ИС через единый узел. Данные решения были апробированы при реализации проекта интеграции разнородных информационных ресурсов органов внутренних дел в единое информационное пространство и изложены авторами в работе [3]. Архитектуру предлагаемого решения можно представить следующим образом:

1. Для интеграции компонентов ИС создается специальный интеграционный слой, предназначенный для реализации взаимодействия между всеми абонентами[1] системы и всеми ее сервисами-поставщиками, представленными как компонентами данной ИС, так и сервисами-поставщиками внешних ИС.

2. Интеграционный слой представляется в виде совокупности взаимодействующих друг с другом узлов двух типов — узлов взаимодействия с абонентами ИС и узлов взаимодействия с сервисами-поставщиками ИС.

3. В интеграционном слое осуществляется публикация ресурсов ИС в виде заявок на выполнение определенных операций и действий (коды, описания заявок, описание параметров, схемы представления результатов исполнения заявок и т. п.), доступных для абонентов интегрируемой системы и исполняемых в конечном счете конкретными сервисами-поставщиками.

4. Для взаимодействия с абонентами ИС определяется единая (возможно распределенная) точка доступа, реализованная в виде универсального сервиса взаимодействия (УСВ), обеспечивающего прием на исполнение от абонентов ИС всех опубликованных в интеграционном слое заявок.

5. Универсальные сервисы взаимодействия размещаются в узлах взаимодействия с абонентами системы, где осуществляется предварительная обработка заявок, в том числе аутентификация, авторизация абонентов, проверка наличия у них прав на выполнение конкретных заявок, формально-логический контроль корректности параметров заявок. Кроме того, в узле взаимодействия с абонентами ведется сбор данных для мониторинга процессов взаимодействия ИС с абонентами системы, осуществляется маршрутизация заявок и их передача на исполнение узлам взаимодействия с соответствующими сервисами-поставщиками.

[1] Под абонентами системы здесь и далее будем подразумевать компоненты, в составе которых реализованы сервисы-потребители, взаимодействующие с интеграционным слоем данной ИС.

6. Узлы взаимодействия с сервисами-поставщиками осуществляют преобразование параметров заявок из универсального формата, в котором они поступают от абонентов, в форматы вызовов соответствующих методов сервисов-поставщиков, организуют очереди сообщений к сервисам-поставщикам, выполняют вызовы и переповторы вызовов сервисов-поставщиков, ведут протоколирование и мониторинг взаимодействия с сервисами-поставщиками ИС.

7. Узлы взаимодействия с сервисами-поставщиками внешних ИС осуществляют, помимо всего прочего, адаптацию требований безопасности данной ИС к условиям их применения во внешних ИС и, соблюдая требования внешних ИС по взаимодействию, передают запросы на исполнение сервисам-поставщикам этих ИС.

Все параметры УСВ разделены на три категории: параметры взаимодействия абонента и УСВ, параметр структурированных данных заявки, параметр неструктурированных данных и документов заявки.

Параметры взаимодействия не зависят от конкретной передаваемой на обработку заявки и предназначены для описания цели, сторон и режима взаимодействия. В состав параметров этой группы входят: код абонента, код заявки, режим взаимодействия абонента с УСВ (синхронный, асинхронный, комбинированный), временны́е ограничения на время исполнения заявки (timeout) и другие параметры.

Параметр структурированных данных содержит сведения о конкретной заявке в виде XML-структуры произвольной конфигурации. В этом параметре содержатся скалярные данные и строки, а ограничения на его представление (XSD-схема) заносятся в реестр заявок ИС при регистрации в нем каждой конкретной заявки. Эти сведения используются УСВ для контроля корректности представленных сервису данных. При необходимости в реестр заявок может заноситься дополнительная информация, необходимая для конвертации структурированных данных из универсального формата УСВ в формат параметров конкретного сервиса-поставщика.

Параметр с неструктурированными данными и документами также относится к переданной на обработку заявке, однако предназначен для передачи УСВ данных больших объемов — графики, текстовых документов и пр. Выделение этих данных в отдельный параметр позволяет при необходимости оптимизировать взаимодействие веб-сервиса абонента и УСВ, организовав их передачу в потоковом режиме, когда их представление в теле SOAP-сообщения основного сеанса взаимодействия становится неприемлемым.

На рис. 3 представлена схема взаимодействия компонентов интеграционного слоя в процессе приема заявок от абонентов ИС, их обработки и исполнения.

В процессе обработки заявки в интеграционном слое осуществляются следующие действия:

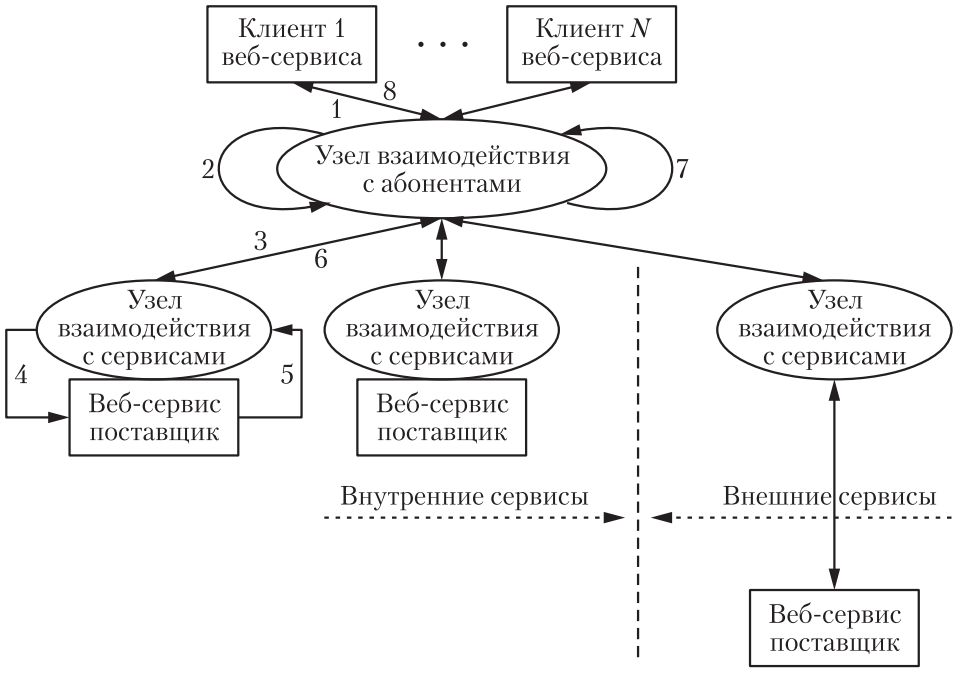


Рис. 3 Взаимодействие компонентов интеграционного слоя

Шаг 1. Клиент отсылает запрос в УСВ интеграционного слоя.

Шаг 2. Выполняется аутентификация, авторизация абонента, проверяется наличие у него прав на выполнение поданной заявки, осуществляется формально-логический контроль корректности параметров заявки.

Шаг 3. Осуществляется маршрутизация заявки и ее отправка на выполнение узлу взаимодействия с соответствующим сервисом-поставщиком.

Шаг 4. Выполняется преобразование параметров заявки из универсального формата в формат вызова соответствующего метода сервиса-поставщика, организуются очереди сообщений к сервисам-поставщикам, выполняются вызовы и переповторы вызовов сервисов-поставщиков, ведется протоколирование и мониторинг взаимодействия ИС с сервисами-поставщиками.

Шаг 5. Веб-сервис-поставщик обрабатывает запрос и возвращает ответ.

Шаг 6. Выполняется формально-логическая проверка корректности полученного ответа.

Шаг 7. Осуществляется протоколирование по результатам получения ответа.

Шаг 8. Ответ возвращается клиенту, инициировавшему данный запрос.

Для реализации большей части функций узлов интегрирующего слоя могут быть использованы различные интеграционные продукты. Возможный вариант реализации интеграционного слоя с использованием продуктов компании Oracle приведен на рис. 4. С помощью агентов WSM реализуются политики доступа (дешифрование, аутентификация, авторизация, мониторинг и протоколирование) при взаимодействии как с клиентами веб-сервисов (абонентами ИС), так и между узлами интеграционного слоя. Маршрутизация, передача заявок между узлами, преобразование универсальной формы параметров заявок в вызовы методов сервисов-поставщиков, формирование очередей сообщений и организация взаимодействия с сервисами-поставщиками может быть реализована с использованием функционала Enterprise Service Bus (ESB).

Следует отметить некоторые важные свойства рассматриваемого решения, а именно: широкий диапазон масштабирования, независимость протокола взаимодействия абонентов с системой от протоколов взаимодействия сервисов-поставщиков (преобразование протоколов взаимодействия), гибкость в организации видов доступа (синхронный/асинхронный) абонентов к системе.

В представленном на рис. 4 решении показан один узел взаимодействия с абонентами и несколько узлов взаимодействия с сервисами-поставщиками. В случае применения решения для интеграции небольших систем уровня предприятия с низкой интенсивностью заявок от клиентов функционал узлов обоих типов может быть объединен в одном узле, обслуживающем всех абонентов системы и организующем взаимодействие с несколькими сервисами-поставщиками ИС. При более высокой интенсивности запросов в интеграционном слое может быть развернуто произвольное количество абонентских узлов. Распределение нагрузки между абонентскими узлами может осуществляться статически, путем «прикрепления» абонентов к соответствующим абонентским узлам, или динамически, при размещении абонентских узлов после балансировщика нагрузки. При создании крупных, территориально распределенных систем абонентские узлы могут устанавливаться вне защищенной зоны предприятия в различных географических точках, максимально приближенных к группам абонентов системы. Это не нарушает требования безопасности, так как для организации взаимодействия абонентских узлов и узлов взаимодействия с сервисами-поставщиками используются защищенные протоколы HTTPS (HyperText Transfer Protocol Secured) (SSL/TLS — Secure Socket Layer / Transport Layer Security).

Сервисы-поставщики ИС могут использовать для организации взаимодействия различные протоколы (HTTP/SOAP (Simple Object Access Protocol), FTP/File, JMS и пр.). В рассматриваемом решении все особенности организации взаимодействия с сервисами-поставщиками локализованы в соответствующих узлах, а полученный результат, как и запрос, всегда предоставляется абоненту через HTTP/SOAP-соединение. Взаимодействие абонента с узлом интеграционного слоя определяется, вообще говоря, независимо от способа доступа к сервису поставщика, исполняющего в конечном счете этот запрос.

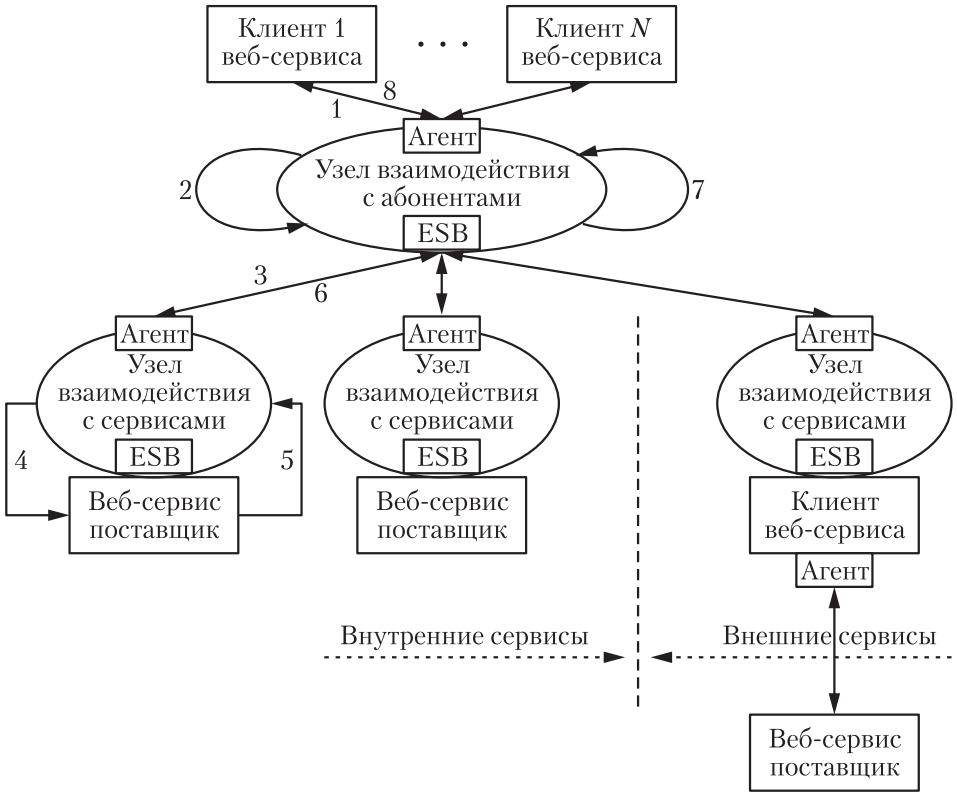


Рис. 4 Реализация интеграционного слоя с использованием продуктов компании Oracle

Как сервисы-поставщики данных, так и абоненты могут реализовывать взаимодействие в синхронном или асинхронном режиме доступа. Единственное ограничение, влияющее на возможность организации синхронного режима взаимодействия абонента с системой, заключается в величине интервала времени, в течение которого ожидается получение ответа от сервиса-поставщика. При асинхронном режиме взаимодействия первая фаза исполнения заявки завершается принятием ее на обработку, абонентский узел выполняет предварительную обработку заявки и возвращает абоненту в качестве результата уникальный идентификатор ожидаемого ответа, а принятая заявка передается сервису-поставщику на исполнение. Во второй фазе абонентский узел, отслеживая момент поступления ответа на заявку, инициирует передачу результатов исполнения заявки инициатору заявки через отдельный тракт, например от клиента сервиса-поставщика к веб-сервису абонента[1].

[1]Тракт взаимодействия узлов интеграционного слоя и абонентов системы для передачи результатов исполнения заявок в асинхронном режиме взаимодействия на рис. 4 не показан.

Итак, подведем некоторые итоги.

1. В результате перехода от публикации сервисов-поставщиков и их методов к публикации заявок на выполнение определенных операций и действий появляется возможность реализации взаимодействия абонентов с ИС через единую точку доступа, которая реализуется в виде универсального веб-сервиса.
2. Независимо от числа используемых в отдельных компонентах ИС различных сервисов-поставщиков в каждом компоненте ИС реализуется только один клиент, осуществляющий взаимодействие с интеграционным слоем путем передачи ему на исполнение любых опубликованных в интеграционном слое заявок.
3. Взаимодействие с каждым сервисом-поставщиком локализовано в одной точке интеграционного слоя — в узле взаимодействия с конкретным сервисом-поставщиком.
4. Взаимодействие абонентов ИС с универсальным сервисом приема заявок может осуществляться в синхронном или асинхронном режиме. Режим доступа задается абонентом вне зависимости от реализации сервиса-поставщика, исполняющего в конечном счете эту заявку.
5. Изменения реализации сервисов-поставщиков или их переход на новые версии, не затрагивающие функциональных интерфейсов, требуют выполнения корректировок только в одной точке (узле взаимодействия с соответствующим сервисом-поставщиком) и не затрагивают реализацию абонентов ИС.
6. Представленное решение предлагает дополнительные (архитектурные) средства масштабирования, обеспечивающие возможность оптимальной загрузки аппаратных средств интеграционного слоя, начиная от небольших ИС уровня предприятия и вплоть до масштабных, территориально распределенных ИС.

4 Текущее состояние разработки и предложения по интеграции прикладных компонентов единой государственной информационной системы в сфере здравоохранения

В начале 2013 г. завершилось создание первой очереди ЕГИСЗ. В конкурсной документации [12] на выполнение работ по созданию второй очереди ЕГИСЗ в качестве основной цели определена эффективная информационная поддержка процесса управления компонентами ЕГИСЗ, разработка новых и развитие функциональности существующих компонентов ЕГИСЗ, а в качестве одной из задач обозначено требование о необходимости разработки подсистемы интеграции прикладных систем (ИПС) ЕГИСЗ.

В состав первой очереди ЕГИСЗ, разрабатываемой в соответствии с концепцией ЕГИСЗ [13], вошли семь основных прикладных ИС федерального сегмента:

(1) централизованная система электронной записи на прием к врачу (федеральная электронная регистратура — ФЭР);

(2) централизованная система ведения интегрированной электронной медицинской карты (ИЭМК);

(3) подсистема административно-хозяйственной деятельности (АХД);

(4) подсистема доступа к нормативно-справочной информации (НСИ) и словарям медицинских терминов;

(5) федеральный сервис «Взаимодействие лекарственных средств» (ВЛС);

(6) информационно-аналитическая система (ИАС);

(7) единая подсистема идентификации, аутентификации и авторизации пользователей (ЕСИАиА) ЕГИСЗ.

Перечисленные компоненты создавались различными компаниями как самостоятельные ИС при отсутствии единого проекта и без принятия согласованных технических решений. Компоненты ЕГИСЗ в процессе реализации своих бизнес-процессов осуществляют информационный обмен как между собой, так и с компонентами внешних ИС, в том числе:

– федеральным регистром медицинского персонала (ФРМП) информационно-аналитической системы Минздрава России;

– единым реестром застрахованных лиц (ЕРЗ) Федерального фонда обязательного медицинского страхования (ФФОМС);

– ИС территориальных ФОМС (ТФОМС) (выгрузка реестров счетов, получение данных об оплате счетов);

– единым порталом государственных услуг (ЕПГУ) посредством системы межведомственного электронного взаимодействия (СМЭВ);

– СМЭВ;

– медицинскими ИС лечебно-профилактических учреждений (ЛПУ).

Взаимодействие компонентов первой очереди ЕГИСЗ организовано следующим образом:

1. Каждый компонент федерального сегмента ЕГИСЗ является сервисом-поставщиком данных и имеет один или несколько программных интерфейсов, позволяющих поместить некую информацию в соответствующую ИС или получить от нее нужные данные.
2. Программные интерфейсы сервисов-поставщиков компонентов ЕГИСЗ реализованы в виде одного или нескольких веб-сервисов.
3. Каждый компонент федерального сегмента ЕГИСЗ, в свою очередь, взаимодействует с несколькими другими компонентами федерального сегмента ЕГИСЗ и с несколькими сторонними ИС (объектами доступа).

4. Перечень объектов доступа для каждого компонента федерального сегмента ЕГИСЗ индивидуален, однако одни и те же объекты доступа присутствуют в перечне объектов доступа нескольких разных компонентов ЕГИСЗ.
5. Программные средства взаимодействия с объектами доступа (сервисами-потребителями) встроены в каждый компонент ЕГИСЗ и, следовательно, дублируют друг друга в различных компонентах ЕГИСЗ.
6. В качестве сервисов-потребителей компонентов федерального сегмента ЕГИСЗ выступают не только другие компоненты федерального сегмента, но и компоненты региональных сегментов ЕГИСЗ, а также медицинских организаций РФ.

Как видно, в первой очереди ЕГИСЗ реализовано взаимодействие компонентов по методу «точка–точка» со всеми рассмотренными ранее проблемами и недостатками. Запланированная к реализации во второй очереди подсистема ИПС должна реализовывать взаимодействие компонентов через «единый узел» и обеспечивать:

– единую точку подключения и информационного взаимодействия подсистем, входящих в состав сегмента централизованных общесистемных компонентов, а также подсистем, входящих в состав сегмента прикладных компонентов регионального уровня при обращении к компонентам федерального;
– первичный контроль информационных сообщений на соответствие форматам;
– контроль корректности и правомерности вызова сервисов федеральных прикладных компонентов;
– маршрутизацию информационных сообщений (адресацию сообщений в соответствующий региональный сегмент);
– мониторинг состояния сервисов федеральных прикладных компонентов;
– сбор статистической информации об объеме и качестве информационного взаимодействия;
– гарантированную доставку неискаженных сообщений от отправителя к получателю в установленные (регламентированные) сроки.

Подсистема ИПС должна быть разработана с поддержкой аппаратной масштабируемости и включать средства настройки, мониторинга и журналирования информационного взаимодействия. В основу проектирования сервиса ИПС должен быть положен принцип максимально возможной централизации доступа к сервисам с учетом распределенного характера взаимодействия в рамках предоставления услуг в сфере здравоохранения и фармакологии.

Предъявляемые к подсистеме ИПС требования практически полностью соответствуют рассмотренным ранее свойствам интеграционного слоя, из чего следует, что рассмотренная архитектура интеграционного слоя может быть с успехом применена при создании подсистемы ИПС ЕГИСЗ.

5 Заключение

В результате анализа различных подходов к интеграции приложений в единую ИС, в том числе с использованием интеграционных продуктов платформы Oracle SOA Suite, выявлены достоинства и недостатки этих решений. Разработано свободное от выявленных недостатков решение по интеграции приложений в единую ИС на базе построения специального интеграционного слоя ИС. Отличительными особенностями данного подхода являются:

- переход от публикации сервисов-поставщиков и их методов (как интерфейса ИС) к публикации заявок на выполнение операций и действий в интегрированной ИС;
- предоставление всем абонентам ИС единого универсального сервиса взаимодействия, осуществляющего прием на выполнение всех зарегистрированных в интеграционном слое заявок;
- локализации взаимодействия с сервисами-поставщиками в одной точке интеграционного слоя.

Показано, что при реализации данного подхода удается существенно сократить затраты на разработку и сопровождение такой ИС, повысить устойчивость системы к изменениям условий ее функционирования, упростить модернизацию и развитие ИС, а также расширить возможности ее масштабирования.

Проведен анализ состояния хода разработки ЕГИСЗ с точки зрения интеграционных решений. Рассмотрены требования к создаваемой подсистеме интеграции прикладных систем ЕГИСЗ. Показано, что изложенное в статье решение по интеграции ИС на базе интеграционного слоя может быть с успехом применено для реализации подсистемы ИПС ЕГИСЗ.

Литература

1. *Илюшин Г. Я., Лиманский В. И.* Реализация механизма типовых запросов в территориально распределенной ведомственной информационной системе // Системы и средства информатики. Доп. вып. — М.: ИПИ РАН, 2009. С. 15–33.
2. *Илюшин Г. Я., Соколов И. А.* Организация управляемого доступа пользователей к разнородным ведомственным информационным ресурсам // Информатика и её применения, 2010. Т. 4. Вып. 1. С. 24–40.
3. *Илюшин Г. Я., Лиманский В. И.* Методы реализации промежуточного слоя на примере систем межведомственного и межрегионального электронного взаимодействия. — М.: ИПИ РАН, 2011. 236 с.
4. *Илюшин Г. Я., Прохоров Н. Л.* Методы и технологии интеграции информационных ресурсов и сервисов на примере медицины // Приборы, 2006. № 3(69). С. 45–52.
5. *Илюшин Г. Я., Лиманский В. И., Володькин А. В.* Особенности создания систем поиска, заказа и бронирования медикаментов с использованием технологий Интернет // Системы и средства информатики, 2003. Вып. 13. С. 329–353.

6. *Илюшин Г. Я., Лиманский В. И., Боков А. М., Подгорнов Ю. Г.* Основные принципы и методы интеграции информационных ресурсов ОВД // Системы и средства информатики. Спец. вып. Научно-технические вопросы построения и развития информационно-телекоммуникационной системы органов внутренних дел. — М.: ИПИ РАН, 2009. С. 34–62.
7. *Шепелявый Д.* Обеспечение безопасности Web-сервисов // Информационная безопасность, 2008. № 1. С. 28–29. http://www.itsec.ru/articles2/Oborandteh/obespe4_bezopasn_web_servisov.
8. Oracle Fusion Middleware: Каталог программных продуктов. http://oracle.axoft.ru/images/Oracle%20Fusion%20Middleware.pdf.
9. Интеграция приложений на платформе Oracle Fusion Middleware. http://www.topsbi.ru/default.asp?artID=1687.
10. О концепции формирования в Российской Федерации электронного правительства до 2010 года: Распоряжение Правительства Российской Федерации № 632-р от 6 мая 2008 г.
11. О единой системе межведомственного электронного взаимодействия: Постановление правительства РФ № 697 от 8 сентября 2010 г.
12. Документация об открытом аукционе в электронной форме на право заключения государственного контракта на выполнение работ по созданию второй очереди прикладных информационных систем в рамках реализации концепции создания единой государственной информационной системы в сфере здравоохранения. Департамент информационных технологий и связи Министерства здравоохранения РФ от 14 мая 2013. http://zakupki.gov.ru/pgz/public/action/orders/info/common_info/show?notificationId=6179244.
13. Концепция создания единой государственной информационной системы в сфере здравоохранения. Приложение к приказу Министерства здравоохранения и социального развития Российской Федерации № 364 от 28 апреля 2011 г.

Поступила в редакцию 20.08.13

METHODS OF CLOUD SERVICE INTEGRATION BY THE EXAMPLE OF HEALTHCARE

G. Ilushin and V. Limansky

Institute of Informatics Problems, Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation

Abstract: The article gives a review of traditional approaches to integration of many independent applications into a unified information system (IS), their merits and demerits. The approaches to integration of applications in the context of the unified IS and requirements for solution of this task based on constructing a special integration layer of an IS are described. The solution is built on the principles of the service-oriented architecture, allows to use modern integration products of different companies, and offers additional architectural tools of ranging. These tools

give an opportunity of optimum loading of software and hardware tools of the integration layer, starting with small IS like enterprises and up to scaled geographically distributed IS. Particularly, the article shows possibility and expediency of using the integration layer mentioned above for implementing the unified state IS in the field of healthcare for serving the task of integration of its components into the unified IS.

Keywords: information systems; service-oriented architecture; web services; integration of components of IS; intermediate layer; unified state information system in the field of healthcare

DOI: 10.14357/08696527140214

References

1. Ilyushin, G. Ya., and V. I. Limanskiy. 2009. Realizatsiya mekhanizma tipovykh zaprosov v territorial'no raspredelennoy vedomstvennoy informatsionnoy sisteme [Typical requests mechanism realization in the departmental distributed information system]. *Sistemy i Sredstva Informatiki — Systems and Means of Informatics*. Additional Issue. 15–33.
2. Ilyushin, G. Ya., and I. A. Sokolov. 2010. Organizatsiya upravlyaemogo dostupa pol'zovateley k raznorodnym vedomstvennym informatsionnym resursam [Organization of users' manageable access to heterogeneous departmental resource]. *Informatika i ee Primeneniya — Inform. Appl.* 4(1):24–40.
3. Ilyushin, G. Ya., and V. I. Limanskiy. 2011. *Metody realizatsii promezhutochnogo sloya na primere sistem mezhvedomstvennogo i mezhregional'nogo elektronnogo vzaimodeystviya* [The middleware implementation methods on the example of interdepartmental and interregional electronic interaction systems]. Moscow: IPI RAN. 236 p.
4. Ilyushin, G. Ya., and N. L. Prokhorov. 2006. Metody i tekhnologii integratsii informatsionnykh resursov i servisov na primere meditsiny [Methods and technologies of information resources and services integration on the example of medicine]. *Pribory* [Devices] 3(69):45–52.
5. Ilyushin, G. Ya., V. I. Limanskiy, and A. V. Volod'kin. 2003. Osobennosti sozdaniya sistem poiska, zakaza i bronirovaniya medikamentov s ispol'zovaniem tekhnologiy Internet [Features of search systems creation, the order and booking of medicines with use of the Internet technologies]. *Sistemy i Sredstva Informatiki — Systems and Means of Informatics* 13:329–353.
6. Ilyushin, G. Ya., V. I. Limanskiy, A. M. Bokov, and Yu. G. Podgornov. 2009. Osnovnye printsipy i metody integratsii informatsionnykh resursov OVD [Basic principles and methods of information resources integration of Department of Internal Affairs]. *Sistemy i Sredstva Informatiki — Systems and Means of Informatics.* Special Issue. 34–62.
7. Shepelyavyy, D. 2008. Obespechenie bezopasnosti Web-servisov [Web services security]. *Informatsionnaya Bezopasnost'* [Information Security] 1:28–29.
8. Oracle Fusion Middleware — semeystvo integrirovannykh produktov [Oracle Fusion Middleware — family of the integrated products]. Available at: http://oracle.axoft.ru/images/Oracle%20Fusion%20Middleware.pdf (accessed February 14, 2014).

9. Integratsiya prilozheniy na platforme Oracle Fusion Middleware [Integration of aplication on the Oracle Fusion Middleware platform]. Available at: http://www.topsbi.ru/default.asp?artID=1687 (accessed February 14, 2014).
10. O kontseptsii formirovaniya v Rossiyskoy Federatsii elektronnogo pravitel'stva do 2010 goda [On the concept of formation of the electronic government in Russian Federation till 2010]. Rasporyazhenie Pravitel'stva Rossiyskoy Federatsii ot 6 maya 2008 g. No. 632-r [The Order of the Government of the Russian Federation No. 632-r of May 6, 2008].
11. O edinoy sisteme mezhvedomstvennogo elektronnogo vzaimodeystviya [About uniform interdepartmental electronic interaction system]. Postanovlenie pravitel'stva RF ot 8 sentyabrya 2010 g. No. 697 [The resolution of the Government of the Russian Federation No. 697 of September 8, 2010].
12. Dokumentatsiya ob otkrytom auktsione v elektronnoy forme na pravo zaklyucheniya gosudarstvennogo kontrakta na vypolnenie rabot po sozdaniyu vtoroy ocheredi prikladnykh informatsionnykh sistem v ramkakh realizatsii kontseptsii sozdaniya edinoy gosudarstvennoy informatsionnoy sistemy v sfere zdravookhraneniya [Documentation about open auction in an electronic form on the right of the conclusion of the state contract for performance of work on creation of the second turn of applied information systems within implementation of the concept of creation of uniform state information system in the health care sphere]. Departament informatsionnykh tekhnologiy i svyazi Ministerstva zdravookhraneniya RF ot 14 maya 2013 [Department of information technologies and communication of Ministry of Health of the Russian Federation of May 14, 2013]. Available at: http://zakupki.gov.ru/pgz/public/action/orders/info/common_info/show?notificationId=6179244 (accessed February 14, 2014).
13. Kontseptsiya sozdaniya edinoy gosudarstvennoy informatsionnoy sistemy v sfere zdravookhraneniya [The concept of uniform state information system creation in the health care sphere]. Prilozhenie k prikazu Ministerstva zdravookhraneniya i sotsial'nogo razvitiya Rossiyskoy Federatsii ot 28 aprelya 2011 No. 364 [The annex to the order of the Ministry of Health and Social Development of the Russian Federation No. 364 of April 28, 2011].

Received August 20, 2013

Contributors

Ilyushin Gennadiy Y. (b. 1947) — Candidate of Science (PhD) in technology, Head of Laboratory, Institute of Informatics Problems, Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; ilushin@atik.ru

Limansky Valery I. (b. 1952) — Candidate of Science (PhD) in technology, senior scientist, Institute of Information Problems, Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333,Moscow, Russian Federation; vlimansky@ipiran.ru

ПРИМЕНЕНИЕ ВЕБ-РЕСУРСОВ СИСТЕМЫ ЗНАНИЙ ИНФОРМАТИКИ СИНФ В УЧЕБНОМ ПРОЦЕССЕ

Б. Н. Куров[1]

Аннотация: Анализируется опыт применения веб-ресурсов распределенной гипермедийной системы знаний информатики СИНФ при обучении студентов. В состав СИНФ входит журнал «Информатика: S-моделирование» и энциклопедия информатики «Инфопедия». В журнале публикуются монографии, статьи, учебные материалы, рецензии, комментарии и различные заметки. В энциклопедии приводятся определения понятий информатики и даются необходимые к ним пояснения. Особенностью СИНФ является постоянное расширение системы знаний путем увеличения объема доступных научных и образовательных ресурсов. Показано, что используемые в учебном процессе веб-ресурсы системы знаний информатики СИНФ обеспечивают повышение качества обучения студентов. Этому также способствует применяемая методика обучения, основанная на сочетании аудиторных занятий с дистанционными (видеозанятия, удобная для студентов skype-связь с преподавателем).

Ключевые слова: веб-ресурсы; система знаний информатики; журнал «Информатика: S-моделирование»; энциклопедия информатики «Инфопедия»; учебный процесс

DOI: 10.14357/08696527140215

1 Введение

Существующие в настоящее время образовательные интернет-ресурсы достаточно разнообразны: образовательные курсы (адресуемые широкому спектру обучающихся), энциклопедии, библиотечные системы, словари, переводчики и др.

Среди образовательных курсов наиболее продвинутыми являются курсы, создаваемые в рамках проекта edX [1], который разработан Массачусетским технологическим институтом (МТИ) и Гарвардским университетом. Открытая программная платформа IMT, на которой строятся курсы, предусматривает видеозанятия, встроенные опросы, мгновенную обратную связь с преподавателями, онлайновые лаборатории и обучение в темпе, предпочитаемом студентом. Эта платформа разработана МТИ (в основном, на языке Python). Обеспечивается свободный доступ разработчиков курсов к IMT. Содержание курсов имеет

[1]Институт проблем информатики Российской академии наук, bnkurov@yandex.ru

очень высокий уровень, поскольку для их создания привлекаются лучшие преподаватели вузов из различных стран (в России, например, из Физтеха). Все онлайн-курсы распространяются в Интернете бесплатно.

Из дистанционных курсов можно также отметить платные подготовительные курсы Учебного центра факультета ВМК МГУ, ориентированные на старшеклассников [2]. Внимания заслуживает их качественное наполнение (теоретический материал, задания на самостоятельную работу, тесты).

Популярными веб-ресурсами являются различные универсальные энциклопедии. Например, платная Encyclopedia Britannica [3] и набирающая популярность бесплатная Википедия. Encyclopedia Britannica превосходит Википедию как по глубине, так и по широте охватываемого материала. Каждая из перечисленных энциклопедий насчитывает миллионы пользователей из разных стран мира.

Особенностью рассматриваемых в статье образовательных веб-ресурсов системы знаний информатики СИНФ является то, что они ориентированы на конкретную область знаний — информатику, предметом которой является символьное моделирование произвольных объектов в человеко-машинной среде (s-моделирование) [4].

Веб-ресурсы СИНФ используются в процессе обучения студентов и магистрантов факультета информационных технологий МИРЭА в рамках дисциплин «Символьное моделирование в информатике», «Теория s-моделирования и ее применение» (читаемых профессором В. Д. Ильиным) и «Алгоритмизация и решение задач управления в s-среде» (АРЗУ) (преподаватель — доцент Б. Н. Куров). Существенную часть содержания каждой из дисциплин составляют разделы теории s-моделирования, авторами которой являются В. Д. Ильин и А. В. Ильин [4].

2 СИНФ как научно-образовательные ресурсы

Система знаний информатики СИНФ ориентирована на исследователей, разработчиков информационных технологий, преподавателей вузов, аспирантов и студентов профильных специальностей. СИНФ — это тип электронного издания, в котором публикуемые материалы представлены в форме гипермедийных документов; кроме того, поддерживается возможность создания презентаций и видеосообщений.

Сайт гипермедийного научного издания СИНФ [5] имеет набор сервисов, необходимых для регистрации предполагаемых авторов, представления и систематизации материалов.

Для публикации материалов достаточно, чтобы они соответствовали тематике и были оформлены в соответствии с требованиями СИНФ.

Важной особенностью СИНФ является ежегодное расширение системы знаний путем увеличения объема доступных научных и образовательных ресур-

сов [6–9]. Между тематически близкими ресурсами существует преемственность: новые материалы дополняют ранее опубликованные.

Ядром СИНФ являются журнал «Информатика: S-моделирование» и энциклопедия информатики «Инфопедия».

Цель журнала — представление и обсуждение научных результатов, представленных в виде статей и книг, содержащих описание методов решения задач s-моделирования и их применения в разработках информационных технологий, сервисов, программных и аппаратных средств, научных и образовательных ресурсов.

Инфопедия служит аккумулятором апробированных знаний, представляющих обновляющийся понятийный аппарат информатики.

Ценность Инфопедии не только в ее составе, но и в том, что пользователь имеет возможность быстро осваивать новые понятия. Наличие гиперссылок на используемые в определениях понятия позволяет пользователю увидеть связи понятий и таким образом прийти к пониманию исходного определения.

3 Об использовании СИНФ в практике преподавания

В практике преподавания дисциплины АРЗУ использовались веб-ресурсы СИНФ, содержащиеся как в журнале «Информатика: S-моделирование», так и в энциклопедии информатики «Инфопедия».

Например, при изучении темы «Сравнение алгоритмов управления» [10] формулировки задач, методы и алгоритмы записывались с использованием языка TSM спецификации объектов s-моделирования [4, 6]. Преимущество такого описания в том, что они приобретают вид, привычный для разработчиков программно реализуемых алгоритмов и программистов.

Для иллюстрации приведем фрагмент описания одной из рассматриваемых задач.

Предположим, что энергосистема (ЭС) включает n электростанций, экономичность работы которых определяется расходными характеристиками — функциями затрат на топливо $y[j](x[j])$, где $x[j]$ — нагрузка j-й станции. Известна суммарная нагрузка энергосистемы $x[c;]$ (c — помета), включающая потери в электрической сети.

Задача оптимального распределения нагрузок между электростанциями заключается в минимизации затрат:

$$y = \mathrm{sum}[j \ldots n] y[j](x[j]) \to \min$$

при ограничениях, заданных условием баланса в ЭС:

$$\mathrm{sum}[j \ldots n] x[j] - x[c;] = 0$$

и допустимыми изменениями нагрузок станций:

$$x[\min; j] \le x[j] \le x[\max; j]\,.$$

Решение сформулированной задачи сводится к решению системы нелинейных уравнений

$$dy[j]/dx[j] + \lambda = 0\,;$$
$$\mathrm{sum}[j \ldots n]x[j] - x[c;] = 0\,.$$

На практике значения функций $y[j](x[j])$ получают путем натурных испытаний для заданного набора значений $x[j]$, т. е. $y[j](x[j])$ — это табличные функции. В алгоритмах оптимального распределения предусматривается их интерполирование.

Возникает задача сравнения двух алгоритмов, в которых используются различные способы построения интерполяционных функций. В первом из них (алгоритме $A[1]$) строится интерполяционный кубический сплайн по точкам $(x[0], y[0]), (x[1], y[1]), \ldots, (x[N], y[N])$ произвольной расходной характеристики.

Сплайн-функция $S(f, x)$ является многочленом третьей степени на каждом из отрезков $[x[i-1], x[i]]$:

$$S(f,x) = f[i](x) = a[i0] + a[i1] * x + a[i2] * x * *2 + a[i3] * x * *3$$

при $x[i-1] \le x \le x[i]$, $i = 1, \ldots, N$, и удовлетворяет следующим условиям:

$$f[i](x[i]) = y[i]\,; \;\; f[1](x[0]) = y[0]\,;$$
$$f[k; j](x[j]) = f[k; j+1](x[j]), \;\; k = 1, 2; \;\; j = 1, \ldots, N-1\,,$$

где k — порядок производной.

Для отыскания коэффициентов $a[i0]$, $a[i1]$, $a[i2]$ и $a[i3]$ функции $S(f, x)$ методом прогонки решается система линейных уравнений (относительно вторых производных сплайна в заданных точках $z[0], z[1], \ldots, z[N]$).

В конкурирующем алгоритме $A[2]$ строится кубический сплайн «с растяжением», когда вместо линейности вторых производных требуется линейность разности $f[''; i](x) - \alpha * f[i](x)$ на $[x[i-1], x[i]]$ ($''$ — помета, обозначающая вторую производную от f):

$$f[''; i](x) - \alpha * f[i](x) =$$
$$= (z[i-1] - \alpha * y[i-1]) * (x[i] - x)/h[i] + (z[i] - \alpha * y[i]) * (x - x[i-1])/h[i]\,,\; \alpha > 0\,.$$

Решением этого дифференциального уравнения является функция $f(x)$:

$$f[i](x) = z[i-1]/\alpha * \mathrm{sh}(\alpha * *(1/2) * (x[i]-x))/\mathrm{sh}(\alpha * *(1/2) * h[i]) +$$
$$+ (y[i-1] - z[i-1])/\alpha * (x[i]-x)/h[i] +$$
$$+ z[i]/\alpha * \mathrm{sh}(\alpha * *(1/2) * (x - x[i-1]))/\mathrm{sh}(\alpha * *(1/2) * h[i]) +$$
$$+ (y[i] - z[i]/\alpha) * (x[i] - x[i-1])/h[i]\,.$$

Продифференцировав это выражение и приняв во внимание непрерывность первых производных, образуем систему из $(N-1)$ линейного уравнения. Дальнейшие действия аналогичны тем, которые выполняются для кубического сплайна.

Этот пример показывает, что строчная запись разнообразных символьных выражений на языке TSM упрощает переход от постановки задачи к алгоритмизации и программированию.

Другой пример применения веб-ресурсов СИНФ [4, 6] связан с решением различных задач управления при изучении темы «Решение оптимизационных задач организационного управления средствами табличных процессоров». Рассматриваются линейные и нелинейные задачи. На предварительном этапе анализируются конкретные производственные условия — управленческая ситуация. В результате экспертного анализа возникает содержательная постановка задачи: выявляются *управляемые переменные* (УП), оптимальные значения которых должны быть найдены, все *ограничения*, учет которых обязателен при поиске значений УП, формулируются *цель управления* в терминах требований, предъявляемых к значениям УП. Отметим, что результаты анализа, полученные на этом этапе, не формализованы. К тому же набор понятий существенно изменяется при переходе от одной задачи к другой.

Следующий этап — это переход от содержательной постановки к построению задачного конструктивного объекта (s-задачи) [4, 6]. S-задача — это четверка Formul, Rulsys, Alg, Prog, где Formul — постановка; Rulsys — множество систем обязательных и ориентирующих правил решения задачи, поставленных в соответствие Formul; Alg — объединение множеств алгоритмов, каждое из которых соответствует одному элементу из Rulsys; Prog — объединение множеств программ, каждое из которых поставлено в соответствие одному из элементов Alg. Анализ содержательной постановки позволяет определить, к какому типу оптимизационной задачи она относится (линейная, целочисленная линейная, нелинейная и др.). Тип задачи определяет систему понятий s-задачи, которая инвариантна к содержательным постановкам задач этого типа.

Не будем приводить содержательную постановку рассматриваемой далее учебной задачи распределения ресурсов. Сразу же перейдем к Formul, принимая во внимание, что целью является изучение составляющих элементов s-задачи и что одной постановке Formul можно поставить в соответствие множество различных содержательных постановок (такое задание дается студентам в одной из лабораторных работ). Отметим только, что анализ содержательной постановки определяет тип задачи. В данном случае — это задача линейного целочисленного программирования.

Вначале приведем Formul для произвольного числа управляемых переменных и ограничений.

Formul s-задачи:

Мет (Память)

Inp (Вход):

Вектор коэффициентов целевой функции: $c[j = 1, \ldots, n]$.
Матрица коэффициентов ограничений: $a[i = 1, \ldots, m,\ j = 1, \ldots, n]$.
Вектор правых частей ограничений: $b[i = 1, \ldots, m]$.

Out (Выход): вектор $x[\max; j = 1, \ldots, n]$.

Правило rul максимизации по $x[j = 1, \ldots, n]$ целевой функции $c[j = 1, \ldots, n] * x[j = 1, \ldots, n]$ при ограничениях $a[i = 1, \ldots, m, j = 1, \ldots, n] * x[j = 1, \ldots, n] \le b[i = 1, \ldots, m]$ и $x[j = 1, \ldots, n] : \mathrm{elem}\, N$[1] имеет следующий вид:

$$\max[x[j = 1, \ldots, n] ::$$
$$:: (a[i = 1, \ldots, m, j = 1, \ldots, n] * x[j = 1, \ldots, n] \le b[i = 1, \ldots, m]) ::$$
$$:: \ x[j = 1, \ldots, n] : \mathrm{elem}\ N](c[j = 1, \ldots, n] * x[j = 1, \ldots, n])\,.$$

Заметим, что введенная система понятий уже не отражает экономического смысла управленческой ситуации распределения ресурсов, а оперирует чисто алгебраическими понятиями. Это важное обстоятельство позволяет использовать в дальнейшем аналогичные формулировки и в других управленческих ситуациях (транспортные задачи, задачи составления смесей, задачи назначений и др.). Однако при анализе результатов решения нужно будет возвращаться к содержательной постановке задачи.

Теперь снабдим Formul данными, извлеченными из содержательной постановки. Имеем две целочисленные управляемые переменные (виды изделий) и четыре производственных ограничения. Первые два ограничения (в нижеследующей формулировке s-задачи) отражают расходы двух видов ресурсов, следующие два — регламентируют минимальное количество выпускаемых изделий, а последнее отражает факт неделимости изделий. Цель управления — максимизировать доход (на заданном отрезке времени), если известен доход от реализации единицы каждого вида изделия.

Formul s-задачи (с данными):

Мет (Память)

Inp (Вход):

Вектор коэффициентов целевой функции: $c[j = 1, 2] = c[16; 60]$.
Матрица коэффициентов ограничений: $a[1; 5; 3{,}5; 10; -1; 0; 0; -1]$.
Вектор правых частей ограничений: $b[80; 180; -10; -12]$.

[1] Здесь N — обозначение множества чисел 0, 1, 2, . . .

Out (Выход):
$x[\max; j = 1, 2]$.

Правило rul максимизации по $x[j = 1, 2]$ целевой функции

$$16 * x[1] + 60 * x[2]$$

при ограничениях

$$x[1] + 5 * x[2] \le 80;\ \ 3{,}5 * x[1] + 10 * x[2] \le 180\,;$$
$$-x[1] \le -10;\ \ -x[2] \le -12\,;$$
$$x[1], x[2] : \text{elem } N$$

запишется так:

$$\max[x[1, 2] :: [(x[1]+5*x[2] \le 80;\ \ 3{,}5*x[1]+10*x[2] \le 180;\ \ -x[1] \le -10;$$
$$-x[2] \le -12) :: x[1], x[2] :\ \text{elem } N](16 * x[1] + 60 * x[2])\,.$$

Решение сформулированной s-задачи:

$$x[\max; 1] = 20\,;\ \ x[\max; 2] = 12\,.$$

Оно было найдено средствами Excel 2007 (надстройка «Поиск решения», где реализован метод ветвей и границ).

Приведенный числовой пример позволяет дать геометрическую интерпретацию алгебраической формулировки Formul и шагов алгоритма (что облегчает студентам понимание материала). Для этого устанавливается соответствие между алгебраическими и геометрическими системами понятий (неравенства — полуплоскости, градиент целевой функции — вектор, направленный из начала координат в точку $c[j = 1, 2]$ и т. д). Поскольку на первом шаге алгоритма, реализующего метод ветвей и границ, снимается условие целочисленности решения, параллельно дается геометрическая интерпретация алгоритма симплекс-метода. Геометрическая интерпретация алгоритмов полезна и при обсуждении целесообразности округления целочисленных решений, оценки влияния погрешностей данных на результат решения. Все это говорит о пользе конструирования различных систем понятий и установления соответствия между ними.

В рассмотренной s-задаче цепочка составляющих Formul $\rightarrow$ Rulsys $\rightarrow$ $\rightarrow$ Alg $\rightarrow$ Prog не имела ветвлений. В более сложной лабораторной работе (предлагаемой магистрантам) наряду с применением метода ветвей и границ требовалось решить эту задачу и методом Гомори, самостоятельно разработав программу на выбранном языке программирования. Таким образом, постановке задачи Formul соответствовали два элемента Rulsys. В результате сравнивались решения, полученные и с помощью Excel, и по разработанной программе. Это позволяло оценить целесообразность применения того или иного метода при разных входных данных.

Таким образом, использование задачного конструктивного объекта позволяло наиболее полно представить процесс решения рассматриваемых задач. Отметим также, что при изучении всех тем приходилось часто обращаться к определениям понятий *s*-моделирования, представленным в Инфопедии.

В дополнение к содержанию дисциплины АРЗУ для углубленного изучения постановок задач распределения ресурсов, методов и алгоритмов их решения студентам рекомендовались разделы монографии [8]. В монографии, в частности, предлагается оригинальная постановка задачи распределения ресурсов. Постановка задачи ориентирована на режим вычислительного эксперимента. Разработанные метод и алгоритм целевого перемещения решения [11] позволяют эксперту-планировщику находить решения, удовлетворяющие требованиям реализуемости и эффективности (когда традиционные оптимизационные методы и алгоритмы бессильны). Под эффективным решением понимается не обязательно экстремальное решение. Понятие об эффективности, зависящее и от конкретных ограничений по ресурсам, и от применяемых показателей экономичности решений, формируется экспертом-планировщиком.

Также обращалось внимание студентов на современный подход к программной реализации предложенных алгоритмов в составе интернет-сервисов. Использование интернет-сервисов воплощает концепцию SaaS (Software as a Service). Алгоритмы вычислений реализуются в серверном приложении (сервисе), а графический интерфейс пользователя — в клиентском. В клиентском приложении вводятся пользовательские данные и отправляются сервису через Интернет в теле запроса на расчеты. Сервис, получив запрос, выполняет расчеты, а результаты отправляет клиентскому приложению.

Постановка задачи, разработанные методы решения и программно воплощенные алгоритмы существенно расширяют арсенал вычислительных средств решения линейных задач распределения ресурсов.

Последующие обсуждения со студентами материалов монографии [8] подтвердили ее безусловную полезность.

4 Организация учебного процесса

Методика преподавания дисциплин совмещает аудиторную и дистанционную форму обучения. На первых занятиях, проводимых в аудитории, студенты получают подробные сведения о содержании дисциплин, составе и сроках выполнения лабораторных работ и курсового проекта. Характеризуются веб-ресурсы СИНФ, которые следует использовать при освоении дисциплин. Большинство следующих занятий проводится дистанционно в режиме онлайн с использованием Skype.

Дисциплины разбиты на темы (12–15 тем). На каждом занятии (которые, как правило, проходят раз в неделю) студентам разъясняется содержание очередной темы, рекомендуются материалы журнала «Информатика: S-моделирование» и статьи энциклопедии «Инфопедия», которые необходимо изучить. Особенность

групп студентов-старшекурсников и магистрантов — их немногочисленность (не более 10 человек). Это позволяет преподавателю на очередном skype-занятии в режиме групповой связи выяснять, насколько студенты освоили материалы темы и готовы ли они к выполнению лабораторных работ. В конце каждого занятия студентам сообщается очередная тема и повторяются все методические приемы, которые использовались применительно к предыдущей теме.

Лабораторные работы играют роль тестов. Задания на лабораторные работы студенты получают в виде файлов (через Skype или по электронной почте). Ответы на вопросы по теоретическому материалу и лабораторным работам студенты также получают через Skype. Оперативно разобраться с возникшими вопросами помогает режим «демонстрации экрана», когда преподаватель показывает студентам рабочий стол и выполняет действия, помогающие им преодолеть затруднения.

Особенность методики преподавания состоит в том, что студенты между занятиями в дополнительные дни в фиксированные интервалы времени могут выйти на skype-связь с преподавателем для выяснения возникших вопросов по теоретическому материалу и лабораторным работам.

По результатам собеседований и материалам выполненных лабораторных работ студент два раза в месяц получает текущую оценку успеваемости. Экзамен студенты сдают в аудитории.

Многолетняя практика преподавания позволяет заключить [12], что так организованное взаимодействие преподавателя со студентами вовремя устраняет возникающие у студентов проблемы и обеспечивает успешное освоение дисциплин в заданные сроки.

5 Заключение

Используемые в учебном процессе веб-ресурсы системы знаний информатики СИНФ обеспечивают повышение качества обучения студентов и магистрантов факультета информационных технологий МИРЭА. Повышению эффективности образовательного процесса способствует и применяемая методика обучения, основанная на сочетании аудиторного и дистанционного обучения.

Литература

1. EdX — совместный проект MIT и Гарвардского университета. https://www.edx.org.
2. Дистанционные подготовительные курсы факультета ВМК МГУ. http://ecmc.cs.msu.ru.
3. Encyclopedia Britannica. http://www.britannica.com.
4. *Ильин А. В., Ильин В. Д.* Основы теории S-моделирования. — М.: ИПИ РАН, 2009. [Электронный ресурс.]
5. Система знаний информатики СИНФ. http://s-modeling.com.

6. *Ильин В. Д.* Система порождения программ. Версия 2013 г. — М.: ИПИ РАН, 2013. 142 с. [Электронный ресурс.]
7. *Ильин А. В., Ильин В. Д.* Информатизация управления статусным соперничеством. — М.: ИПИ РАН, 2013. 128 с. [Электронный ресурс.]
8. *Ильин А. В.* Экспертное планирование ресурсов. — М.: ИПИ РАН, 2013. 58 с. [Электронный ресурс.]
9. *Ильин А. В., Ильин В. Д.* Научно-образовательные веб-ресурсы. S-моделирование. — М.: ИПИ РАН, 2013. 112 с. [Электронный ресурс.]
10. *Куров Б. Н.* Сравнение эффективности алгоритмов управления с учетом точности данных и реализации решений // Управление большими системами, 2011. Вып. 34. С. 279–291.
11. *Ilyin A. V., Ilyin V. D.* The technology of interactive resource allocation in accordance with the customizable system of rules // Appl. Math. Sci., 2013. Vol. 7. No. 143. P. 7105–7111. http://dx.doi.org/10.12988/ams.2013.311649.
12. *Куров Б. Н.* Опыт применения электронных ресурсов в обучении студентов МИРЭА // Информационные технологии в образовании: Сб. трудов XXIII Междунар. конф.-выставки. — М.: ВМК МГУ, 2013. Ч. II. С. 48–49.

Поступила в редакцию 28.02.14

APPLICATION OF WEB RESOURCES OF THE SINF INFORMATICS KNOWLEDGE SYSTEM IN EDUCATIONAL PROCESS

B. N. Kurov

Institute of Informatics Problems, Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation

Abstract: The article analyzes the experience of using web resources of the distributed hypermedia system of informatics knowledge (SINF) for teaching students. SINF consists of the journal "INFORMATICS: S-modeling" and the informatics encyclopedia "INFOPEDIYA." The journal publishes monographs, articles, tutorials, reviews, comments, and various notes. The encyclopedia provides definitions of informatics concepts and gives necessary explanations. The feature of SINF is continuous expansion of knowledge system by increasing the amount of available scientific and educational resources. It is shown that the SINF web resources, which are used in educational process, contribute to enhancement of its quality. The quality of teaching is also improved by combination of classroom and remote studies (video lectures, Skype sessions with lecturer).

Keywords: web resources; informatics knowledge system; journal "INFORMATICS: S-modeling," Encyclopedia of informatics "INFOPEDIYA," educational process

DOI: 10.14357/08696527140215

References

1. EdX — a joint project of MIT and Harvard University. Available at: https://www.edx.org/ (accessed March 20, 2014).
2. Distantsionnye kursy fakul'teta VMK MGU im. M. V. Lomonosova [Distance learning courses of the Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University]. Available at: http://ecmc.cs.msu.ru/ (accessed March 20, 2014).
3. Encyclopedia Britannica. Available at: http://www.britannica.com/ (accessed March 20, 2014).
4. Ilyin, A. V., and V. D. Ilyin. 2009. *Osnovy teorii s-modelirovaniya* [*Basics of the theory of s-modeling*]. Moscow: Institute of Informatics Problems of RAS. 143 p. Available at: http://smodeling.files.wordpress.com/2010/01/basics-theory-sm-20092.pdf (accessed March 20, 2014).
5. Sistema znaniy informatiki SINF [The informatics knowledge system SINF]. Available at: http://s-modeling.com (accessed March 20, 2014).
6. Ilyin, V. D. 2013. *Sistema porozhdeniya programm. Versiya 2013 g.* [*The system of program generating. Version 2013*]. Moscow: Institute of Informatics Problems of RAS. 142 p. Available at: http://smodeling.files.wordpress.com/2013/12/vd-ilyin-spp-2013.pdf (accessed March 20, 2014).
7. Ilyin, A. V., and V. D. Ilyin. 2013. *Informatizatsiya upravleniya statusnym sopernichestvom* [*Informatization of status rivalry governance*]. Moscow: Institute of Informatics Problems of RAS. 152 p. Available at: http://smodeling.files.wordpress.com/2014/02/avivdi-iuss-for-sites.pdf (accessed March 20, 2014).
8. Ilyin, A. V. 2013. *Ekspertnoe planirovanie resursov* [*Expert resource planning*]. Moscow: Institute of Informatics Problems of RAS. 58 p. Available at: http://smodeling.files.wordpress.com/2013/12/av-ilyin-epr-2013.pdf (accessed March 20, 2014).
9. Ilyin, A. V., and V. D. Ilyin. 2013. *Nauchno-obrazovatel'nye veb-resursy. S-modelirovanie* [*Scientific-educational Web-resources. S-modeling*]. Moscow: Institute of Informatics Problems of RAS. 112 p. Available at: http://smodeling.files.wordpress.com/2013/12/nowrsm-2013.pdf (accessed March 20, 2014).
10. Kurov, B. N. 2011. Sravnenie effektivnosti algoritmov upravleniya s uchetom tochnosti dannykh i realizatsii resheniy [Comparison of control algorithms efficiency taking into account data accuracy and the solutions implementation]. *Upravlenie Bol'shimi Sistemami* [*Large-Scale Systems Control*] 34:279–291. Available at: http://www.mathnet.ru/links/64315d50eedbfcea856df9db8caa9e18/ubs563.pdf (accessed March 20, 2014).
11. Ilyin, A. V., and V. D. Ilyin. 2013. The technology of interactive resource allocation in accordance with the customizable system of rules. *Applied Math. Sci.* 7(143):7105–7111. doi: 10.12988/ams.2013.311649.
12. Kurov, B. N. 2013. Opyt primeneniya elektronnykh resursov v obuchenii studentov MIREA [Experience of using electronic resources in teaching students of MIREA].

Tr. 23-y Mezhdunar. Konf. i Vystavki "Informatsionnye Tekhnologii v Obrazovanii" [23rd Conference and Exhibition (International) "Information Technologies in Education" Proceedings]. Moscow. II:48–49.

Received February 28, 2014

Contributor

Kurov Boris N. (b. 1939) — Candidate of Science (PhD) in technology, senior scientist, Institute of Informatics Problems, Russian Academy of Science, 44-2 Vavilov Str., Moscow 119333, Russian Federation; bnkurov@yandex.ru

ОБ АВТОРАХ

Аверина Анастасия Андреевна (р. 1983) — руководитель отдела ООО «ИТС-ЭКСПЕРТ»

Агафонов Егор Сергеевич (р. 1981) — научный сотрудник Института проблем информатики Российской академии наук

Адамович Игорь Михайлович (р. 1934) — кандидат технических наук, заведующий отделом Института проблем информатики Российской академии наук

Алексеенко Андрей Николаевич (р. 1983) — руководитель направления ООО «ИТС-ЭКСПЕРТ»

Андрианова Алефтина Александровна (р. 1991) — магистр техники и технологии Санкт-Петербургского государственного политехнического университета

Белоусов Василий Владимирович (р. 1977) — кандидат технических наук, заведующий сектором Института проблем информатики Российской академии наук

Бородина Александра Валентиновна (р. 1980) — кандидат физико-математических наук, научный сотрудник Института прикладных математических исследований Карельского научного центра Российской академии наук; доцент Петрозаводского государственного университета

Веревкин Геннадий Федорович (р. 1934) — кандидат технических наук, старший научный сотрудник Института проблем информатики Российской академии наук

Волков Олег Игоревич (р. 1964) — ведущий программист Института проблем информатики Российской академии наук

Гай Мария Алексеевна (р. 1983) — старший аналитик ООО «ИТС-ЭКСПЕРТ»

Гумникова Татьяна Сергеевна (р. 1957) — эксперт ОАО «РОСОБОРОНЭКСПОРТ»

Гурьев Дмитрий Константинович (р. 1978) — старший программист ООО «ИТС-ЭКСПЕРТ»

Дулин Сергей Константинович (р. 1950) — доктор технических наук, профессор; главный научный сотрудник Открытого акционерного общества «Научно-исследовательский и проектно-конструкторский институт информатизации, автоматизации и связи на железнодорожном транспорте» (ОАО «НИИАС»); старший научный сотрудник Института проблем информатики Российской академии наук

Дулина Наталья Георгиевна (р. 1947) — кандидат технических наук, старший научный сотрудник Вычислительного центра им. А. А. Дородницына Российской академии наук

Илюшин Геннадий Яковлевич (р. 1947) — кандидат технических наук, заведующий лабораторией Института проблем информатики Российской академии наук

Иткин Иосиф Леонидович (р. 1978) — генеральный директор компании "Exactpro Systems", США, Калифорния, Сан-Рафаэль

Ицыксон Владимир Михайлович (р. 1973) — кандидат технических наук, доцент Санкт-Петербургского государственного политехнического университета

Козеренко Елена Борисовна (р. 1959) — кандидат филологических наук, заведующая лабораторией Института проблем информатики Российской академии наук

Корепанов Эдуард Рудольфович (р. 1966) — кандидат технических наук, заведующий сектором Института проблем информатики Российской академии наук

Куров Борис Николаевич (р. 1939) — кандидат технических наук, старший научный сотрудник Института проблем информатики Российской академии наук

Лиманский Валерий Иванович (р. 1952) — кандидат технических наук, старший научный сотрудник Института проблем информатики Российской академии наук

Лукьянов Геннадий Викторович (р. 1952) — кандидат военных наук, доцент; заведующий сектором Института проблем информатики Российской академии наук

Морозов Евсей Викторович (р. 1947) — доктор физико-математических наук, профессор, ведущий научный сотрудник Института прикладных математических исследований Карельского научного центра Российской академии наук; профессор Петрозаводского государственного университета

Морозов Николай Викторович (р. 1956) — старший научный сотрудник Института проблем информатики Российской академии наук

Морозова Юлия Игоревна (р. 1984) — научный сотрудник Института проблем информатики Российской академии наук

Никишин Дмитрий Александрович (р. 1976) — кандидат технических наук, заведующий сектором Института проблем информатики Российской академии наук

Петрухин Владимир Сергеевич (р. 1949) — старший научный сотрудник Института проблем информатики Российской академии наук

Проценко Павел Александрович (р. 1983) — старший менеджер проектов компании "Exactpro Systems", США, Калифорния, Сан-Рафаэль

Сергеев Игорь Викторович (р. 1965) — кандидат технических наук, заместитель директора Института проблем информатики Российской академии наук

Синицын Игорь Николаевич (р. 1940) — доктор технических наук, профессор, заслуженный деятель науки РФ, заведующий отделом Института проблем информатики Российской академии наук

Стенина Мария Михайловна (р. 1991) — студентка Московского физико-технического института

Степченков Дмитрий Юрьевич (р. 1973) — старший научный сотрудник Института проблем информатики Российской академии наук
Степченков Юрий Афанасьевич (р. 1951) — кандидат технических наук, заведующий отделом Института проблем информатики Российской академии наук
Стрижов Вадим Викторович (р. 1967) — кандидат физико-математических наук, доцент, научный сотрудник Вычислительного центра им. А. А. Дородницына Российской академии наук
Сучков Александр Павлович (р. 1954) — доктор технических наук, ведущий научный сотрудник Института проблем информатики Российской академии наук
Терентьев Александр Александрович (р. 1951) — доктор технических наук, директор по исследованиям ООО «ИТС-ЭКСПЕРТ»
Трифанов Виталий Юрьевич (р. 1988) — старший инженер-программист ООО «Эксперт-Система СЗ»
Цителов Дмитрий Игоревич (р. 1974) — руководитель проекта ООО «Эксперт-Система»
Шаламов Анатолий Степанович (р. 1947) — доктор технических наук, профессор, консультант Института проблем информатики Российской академии наук
Шарнин Михаил Михайлович (р. 1959) — кандидат технических наук, старший научный сотрудник Института проблем информатики Российской академии наук
Шаров Даниил Сергеевич (р. 1991) — программист ООО «ИТС-ЭКСПЕРТ»
Шоргин Всеволод Сергеевич (р. 1978) — кандидат технических наук, старший научный сотрудник Института проблем информатики Российской академии наук

Правила подготовки рукописей статей для публикации в журнале «Системы и средства информатики»

Журнал «Системы и средства информатики» публикует теоретические, обзорные и дискуссионные статьи, посвященные научным исследованиям и разработкам в области информационных технологий.

Журнал издается на русском языке. По специальному решению редколлегии отдельные статьи могут печататься на английском языке.

Тематика журнала охватывает следующие направления:

- информационно-телекоммуникационные системы и средства их построения;
- архитектура и программное обеспечение вычислительных машин, комплексов и сетей;
- методы и средства защиты информации.

1. В журнале печатаются статьи, содержащие результаты, ранее не опубликованные и не предназначенные к одновременной публикации в других изданиях.

   Публикация не должна нарушать закон об авторских правах.

   Направляя рукопись в редакцию, авторы сохраняют все права собственников данной рукописи и при этом передают учредителям и редколлегии неисключительные права на издание статьи на русском языке (или на языке статьи, если он отличен от русского) и на ее распространение в России и за рубежом. Авторы должны предоставить в редакцию письмо в следующей форме:

   Соглашение о передаче права на публикацию:

   «Мы, нижеподписавшиеся, авторы рукописи «. . . », передаем учредителям и редколлегии журнала «Системы и средства информатики» неисключительное право опубликовать данную рукопись статьи на русском языке как в печатной, так и в электронной версиях журнала. Мы подтверждаем, что данная публикация не нарушает авторских прав других лиц или организаций.

   Подписи авторов: (ф. и. о., дата, адрес)».

   Это соглашение может быть предоставлено в бумажном виде или в виде отсканированной копии (с подписями авторов).

   Редколлегия вправе запросить у авторов экспертное заключение о возможности публикации представленной статьи в открытой печати.
2. К статье прилагаются данные автора (авторов) (см. п. 8). При наличии нескольких авторов указывается фамилия автора, ответственного за переписку с редакцией.
3. Редакция журнала осуществляет экспертизу присланных статей в соответствии с принятой в журнале процедурой рецензирования.

   Возвращение рукописи на доработку не означает ее принятия к печати.

   Доработанный вариант с ответом на замечания рецензента необходимо прислать в редакцию.
4. Решение редколлегии о публикации статьи или ее отклонении сообщается авторам.

   Редколлегия может также направить авторам текст рецензии на их статью. Дискуссия по поводу отклоненных статей не ведется.
5. Редактура статей высылается авторам для просмотра. Замечания к редактуре должны быть присланы авторами в кратчайшие сроки.
6. Рукопись предоставляется в электронном виде в форматах MS WORD (.doc или .docx) или LaTeX (.tex), дополнительно — в формате .pdf, на дискете, лазерном диске или электронной почтой. Предоставление бумажной рукописи необязательно.

7. При подготовке рукописи в MS Word рекомендуется использовать следующие настройки.

   Параметры страницы: формат — А4; ориентация — книжная; поля (см): внутри — 2,5, снаружи — 1,5, сверху и снизу — 2, от края до нижнего колонтитула — 1,3.

   Основной текст: стиль — «Обычный», шрифт — Times New Roman, размер — 14 пунктов, абзацный отступ — 0,5 см, 1,5 интервала, выравнивание — по ширине.

   Рекомендуемый объем рукописи — не свыше 20 страниц указанного формата.

   Сокращения слов, помимо стандартных, не допускаются. Допускается минимальное количество аббревиатур.

   Все страницы рукописи нумеруются.

   Шаблоны примеров оформления представлены в Интернете:

   http://www.ipiran.ru/publications/collected/template.doc

8. Статья должна содержать следующую информацию на ***русском и английском языках***:

   - Название статьи.
   - Ф.И.О. авторов, на английском можно только имя и фамилию.
   - Место работы, с указанием города и страны и электронного адреса каждого автора.
   - Сведения об авторах, в соответствии с форматом, образцы которого представлены на страницах:
     http://www.ipiran.ru/journal/collected/2012_22_02_rus/authors.asp и
     http://www.ipiran.ru/journal/collected/2012_22_02_eng/authors.asp
   - Аннотация (не менее 100 слов на каждом из языков). Аннотация — это краткое резюме работы, которое может публиковаться отдельно. Она является основным источником информации в информационных системах и базах данных. Английская аннотация должна быть оригинальной, может не быть дословным переводом русского текста и должна быть написана хорошим английским языком. В аннотации не должно быть ссылок на литературу и, по возможности, формул.
   - Ключевые слова — желательно из принятых в мировой научно-технической литературе тематических тезаурусов. Предложения не могут быть ключевыми словами.

9. Требования к спискам литературы.
   Ссылки на литературу в тексте статьи нумеруются (в квадратных скобках) и располагаются в каждом из списков литературы в порядке первых упоминаний.

   Списки литературы представляются в двух вариантах:

   (1) **Список литературы к русскоязычной части**. Русские и английские работы — на языке и в алфавите оригинала.

   (2) **References**. Русские работы и работы на других языках — в латинской транслитерации с переводом на английский язык; английские работы и работы на других языках — на языке оригинала.

Необходимо для составления списка “References” пользоваться размещенной на сайте http://www.translit.ru/ бесплатной программой транслитерации русского текста в латиницу, при этом в закладке «варианты. . . » следует выбрать опцию BNG.

Список литературы “References” приводится полностью отдельным блоком, повторяя все позиции из списка литературы к русскоязычной части, независимо от того, имеются или нет в нем иностранные источники. Если в списке литературы к русскоязычной части есть ссылки на иностранные публикации, набранные латиницей, они полностью повторяются в списке “References”.

Примеры ссылок на различные виды публикаций в списке “References”:

Описание статьи из журнала:

Zhang, Z., and D. Zhu. 2008. Experimental research on the localized electrochemical micromachining. *Rus. J. Electrochem.* 44(8):926–930. doi:10.1134/S1023193508080077.

Описание статьи из электронного журнала:

Swaminathan, V., E. Lepkoswka-White, and B. P. Rao. 1999. Browsers or buyers in cyberspace? An investigation of electronic factors influencing electronic exchange. *JCMC* 5(2). Available at: http://www.ascusc.org/jcmc/vol5/issue2/ (accessed April 28, 2011).

Описание материалов конференций:

Usmanov, T. S., A. A. Gusmanov, I. Z. Mullagalin, R. Ju. Muhametshina, A. N. Chervyakova, and A. V. Sveshnikov. 2007. Osobennosti proektirovaniya razrabotki mestorozhdeniy s primeneniem gidrorazryva plasta [Features of the design of field development with the use of hydraulic fracturing]. *Trudy 6-go Mezhdunarodnogo Simpoziuma “Novye resursosberegayushchie tekhnologii nedropol’zovaniya i povysheniya neftegazootdachi”* [6th Symposium (International) “New Energy Saving Subsoil Technologies and the Increasing of the Oil and Gas Impact” Proceedings]. Moscow. 267–272.

Описание книги (монографии, сборники):

Lindorf, L. S., and L. G. Mamikoniants, eds. 1972. *Ekspluatatsiya turbogeneratorov s neposredstvennym okhlazhdeniem* [Operation of turbine generators with direct cooling]. Moscow: Energy Publs. 352 p.

Описание переводной книги (в списке литературы к русскоязычной части необходимо указать: / Пер. с англ. — после названия книги, а в конце ссылки указать оригинал книги в круглых скобках):

1. В русскоязычной части:

   Тимошенко С. П., Янг Д. Х., Уивер У. Колебания в инженерном деле / Пер. с англ. — М.: Машиностроение, 1985. 472 с. (*Timoshenko S. P., Young D. H., Weaver W.* Vibration problems in engineering. — 4th ed. — N.Y.: Wiley, 1974. 521 p.)

2. В англоязычной части:

   Timoshenko, S. P., D. H. Young, and W. Weaver. 1974. *Vibration problems in engineering*. 4th ed. N.Y.: Wiley. 521 p.

Описание неопубликованного документа:

Latypov, A. R., M. M. Khasanov, and V. A. Baikov. 2004. Geology and production (NGT GiD). Certificate on official registration of the computer program No. 2004611198. (In Russian, unpubl.)

Описание интернет-ресурса:

Pravila tsitirovaniya istochnikov [Rules for the citing of sources]. Available at: http://www.scribd.com/doc/1034528/ (accessed February 7, 2011).

Описание диссертации или автореферата диссертации:

Semenov, V. I. 2003. Matematicheskoe modelirovanie plazmy v sisteme kompaktnyy tor [Mathematical modeling of the plasma in the compact torus]. D.Sc. Diss. Moscow. 272 p.

Kozhunova, O. S. 2009. Tekhnologiya razrabotki semanticheskogo slovarya informatsionnogo monitoringa [Technology of development of semantic dictionary of information monitoring system]. PhD Thesis. Moscow: IPI RAN. 23 p.

Описание ГОСТа:

GOST 8.586.5-2005. 2007. Metodika vypolneniya izmereniy. Izmerenie raskhoda i kolichestva zhidkostey i gazov s pomoshch'yu standartnykh suzhayushchikh ustroystv [Method of measurement. Measurement of flow rate and volume of liquids and gases by means of orifice devices]. Moscow: Standardinform Publs. 10 p.

Описание патента:

Bolshakov, M. V., A. V. Kulakov, A. N. Lavrenov, and M. V. Palkin. 2006. Sposob orientirovaniya po krenu letatel'nogo apparata s opticheskoy golovkoy samonavedeniya [The way to orient on the roll of aircraft with optical homing head]. Patent RF No. 2280590.

10. Присланные в редакцию материалы авторам не возвращаются.
11. При отправке файлов по электронной почте просим придерживаться следующих правил:
    - указывать в поле subject (тема) название журнала и фамилию автора;
    - использовать attach (присоединение);
    - в состав электронной версии статьи должны входить: файл, содержащий текст статьи, и файл(ы), содержащий(е) иллюстрации.
12. Журнал «Системы и средства информатики» является некоммерческим изданием. Плата за публикацию не взимается, гонорар авторам не выплачивается.

Адрес редакции журнала «Системы и средства информатики»:

Москва 119333, ул. Вавилова, д. 44, корп. 2, ИПИ РАН

Тел.: +7 (499) 135-86-92 Факс: +7 (495) 930-45-05

e-mail: rust@ipiran.ru (Сейфуль-Мулюков Рустем Бадриевич)

http://www.ipiran.ru/journal/collected

Requirements for manuscripts submitted to Journal "Systems and Means of Informatics"

Journal "Systems and Means of Informatics" publishes theoretical, review, and discussion articles on the research and development in the field of information technology.

The journal is published in Russian. By a special decision of the editorial board, some articles can be published in English.

Topics covered include the following areas:

- information and communication systems and tools of their design;
- architecture and software of computational complexes and networks; and
- methods and tools of information protection.

1. The Journal publishes original articles which have not been published before and are not intended for publication in other editions. An article submitted to the Journal must not violate the Copyright law. Sending the manuscript to the Editorial Board, the authors retain all rights of the owners of the manuscript and transfer the nonexclusive rights to publish the article in Russian (or the language of the article, if not Russian) and its distribution in Russia and abroad to the Founders and the Editorial Board. Authors should submit a letter to the Editorial Board in the following form:

   Agreement on the transfer of rights to publish:

   "*We, the undersigned authors of the manuscript ". . . ," pass to the Founder and the Editorial Board of the Journal "Systems and Means of Informatics" the nonexclusive right to publish the manuscript of the article in Russian (or in English) in both print and electronic versions of the Journal. We affirm that this publication does not violate the Copyright of other persons or organizations.*

   Author(s) signature(s): (name(s), address(es), date)."

   This agreement should be submitted in paper form or in the form of a scanned copy (signed by the authors).

   The Editorial Board has the right to request from the authors an official expert conclusion that the submitted article has no secret data prohibited for publication.

2. A submitted article should be attached with **the data on the author(s)** (see item 8). If there are several authors, the contact person should be indicated who is responsible for correspondence with the Editorial Board and other authors about revisions and final approval of the proofs.
3. The Editorial Board of the Journal examines the article according to the established reviewing procedure. If authors receive their article for correction after reviewing, it does not mean that the article is approved to be published. The corrected article should be sent to the Editorial Board for the subsequent review and approval.
4. The decision on the article publication or its rejection is communicated to the authors. The Editorial Board may also send the reviews on the submitted articles to the authors. Any discussion upon the rejected articles is not possible.
5. The edited articles will be sent to the authors for proofread. The comments of the authors to the edited text of the article should be sent to the Editorial Board as soon as possible.
6. The manuscript of the article should be presented electronically in the MS WORD (.doc or .docx) or LaTeX (.tex) formats, and additionally in the .pdf format. All documents may be sent by e-mail or provided on a CD or diskette. A hard copy submission is not necessary.

7. The recommended typesetting instructions for manuscript.

   Pages parameters: format A4, portrait orientation, document margins (cm): left — 2.5, right — 1.5, above — 2.0, below — 2.0, footer 1.3.

   Text: font —Times New Roman, font size — 14, paragraph indent — 0.5, line spacing — 1.5, justified alignment.

   The recommended manuscript size: not more than 20 pages of the specified format.

   Use only standard abbreviations. Avoid abbreviations in the title and abstract. The full term for which an abbreviation stands should precede its first use in the text unless it is a standard unit of measurement.

   All pages of the manuscript should be numbered.

   The templates for the manuscript typesetting are presented on site:

   http://www.ipiran.ru/publication/collected/template.doc

8. Articles should enclose data both in **Russian and English**:

   - title;
   - author's name and surname;
   - affiliation — organization, its address with ZIP code, city, country, and official e-mail address;
   - data on authors according to the format (see site):
     http://www.ipiran.ru/journal/collected/2012_22_02_rus/authors.asp and
     http://www.ipiran.ru/journal/collected/2012_22_02_eng/authors.asp;
   - abstract (not less than 100 words) both in Russian and in English. Abstract is a short summary of the article that can be published separately. The abstract is the main source of information on the article and it could be included in leading information systems and data bases. The abstract in English has to be an original text and should not be an exact translation of the Russian one. Good English is required. In abstracts, avoid references and formulae.
   - Indexing is performed on the basis of keywords. The use of keywords from the internationally accepted thematic Thesauri is recommended.

     Important! Keywords must not be sentences.

9. References. Russian references have to be presented both in English translation and in Latin transliteration.

   Please take into account the following examples of Russian references appearance:

   Article in journal:

   Zhang, Z., and D. Zhu. 2008. Experimental research on the localized electrochemical micromachining. *Rus. J. Electrochem.* 44(8):926–930. doi:10.1134/S1023193508080077.

   Journal article in electronic format:

   Swaminathan, V., E. Lepkoswka-White, and B. P. Rao. 1999. Browsers or buyers in cyberspace? An investigation of electronic factors influencing electronic exchange. *JCMC* 5(2). Available at: http://www.ascusc.org/jcmc/vol5/issue2/ (accessed April 28, 2011).

Conference proceedings:

Usmanov, T. S., A. A. Gusmanov, I. Z. Mullagalin, R. Ju. Muhametshina, A. N. Chervyakova, and A. V. Sveshnikov. 2007. Osobennosti proektirovaniya razrabotki mestorozhdeniy s primeneniem gidrorazryva plasta [Features of the design of field development with the use of hydraulic fracturing]. *Trudy 6-go Mezhdunarodnogo Simpoziuma "Novye resursosberegayushchie tekhnologii nedropol'zovaniya i povysheniya neftegazootdachi"* [6th Symposium (International) "New Energy Saving Subsoil Technologies and the Increasing of the Oil and Gas Impact" Proceedings]. Moscow. 267–272.

Books and other monographs:

Lindorf, L. S., and L. G. Mamikoniants, eds. 1972. *Ekspluatatsiya turbogeneratorov s neposredstvennym okhlazhdeniem* [Operation of turbine generators with direct cooling]. Moscow: Energy Publs. 352 p.

Dissertation and Thesis:

Kozhunova, O. S. 2009. Tekhnologiya razrabotki semanticheskogo slovarya informatsionnogo monitoringa [Technology of development of semantic dictionary of information monitoring system]. PhD Thesis. Moscow: IPI RAN. 23 p.

State standards and patents:

GOST 8.586.5-2005. 2007. Metodika vypolneniya izmereniy. Izmerenie raskhoda i kolichestva zhidkostey i gazov s pomoshch'yu standartnykh suzhayushchikh ustroystv [Method of measurement. Measurement of flow rate and volume of liquids and gases by means of orifice devices]. M.: Standardinform Publs. 10 p.

Bolshakov, M. V., A. V. Kulakov, A. N. Lavrenov, and M. V. Palkin. 2006. Sposob orientirovaniya po krenu letatel'nogo apparata s opticheskoy golovkoy samonavedeniya [The way to orient on the roll of aircraft with optical homing head]. Patent RF No. 2280590.

References in Latin transcription are presented in the original language.

References in the text are numbered according to the order of their first appearance; the number is placed in square brackets. All items from the reference list should be cited.

10. Manuscripts and additional materials are not returned to Authors by the Editorial Board.
11. Submissions of files by e-mail must include:
    - the journal title and author's name in the "Subject" field;
    - an article and additional materials have to be attached using the "attach" function;
    - an electronic version of the article should contain the file with the text and a separate file with figures.
12. "System and Means of Informatics" journal is not a profit publication. There are no charges for the authors as well as there are no royalties.

Editorial Board address:

IPI RAN, 44, block 2, Vavilov Str., Moscow 119333, Russia

Ph.: +7 (499) 135 86 92, Fax: +7 (495) 930 45 05

e-mail: rust@ipiran.ru (to Prof. Rustem Seyful-Mulyukov)

http://www.ipiran.ru/english/journal_systems.asp

SYSTEMS AND MEANS OF INFORMATICS (SISTEMY I SREDSTVA INFORMATIKI)

SCIENTIFIC JOURNAL

Volume 24 No.2 Year 2014

I N T H I S I S S U E: