



# ИНФОРМАТИКА, ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И УПРАВЛЕНИЕ

УДК 519.2:004.421.5:004.7

**В. Н. ЗАДОРОВНИЙ**Омский государственный  
технический университет

## ПОВЫШЕНИЕ ТОЧНОСТИ GPSS-МОДЕЛЕЙ ПУТЕМ ПРИМЕНЕНИЯ ГЕНЕРАТОРА СЛУЧАЙНЫХ ЧИСЕЛ «ВИХРЬ МЕРСЕННА»

Рассматриваются подходы к применению в GPSS World внешнего генератора случайных чисел. Приводятся примеры применения генератора, основанного на алгоритме «Вихрь Мерсенна», для моделирования систем с очередями. Демонстрируются появляющиеся при этом возможности существенного повышения точности результатов моделирования.

**Ключевые слова:** имитационное моделирование, генератор случайных чисел, системы массового обслуживания, распределения с тяжелыми хвостами.

**1. Введение.** Несмотря на широкое распространение множества разнообразных систем имитационного моделирования, система GPSS World [1] до сих пор является одной из самых популярных. Для многих задач моделирования язык GPSS является идеальным средством описания моделей и выполнения имитационных экспериментов. Привлекают также достаточно высокие скорость и точность выполнения большинства вычислительных процессов, реализуемых на GPSS. Вместе с тем

естественная прежде упрощенная (по современным меркам) реализация генераторов случайных чисел (ГСЧ), встроенных в GPSS World, становится в настоящее время препятствием для повышения точности результатов моделирования за счет увеличения объемов выборок до больших размеров, вполне достижимых на современных персональных компьютерах (ПК) средней мощности.

Как показывают эксперименты (которые трудно воспроизвести и проверить), ГСЧ класса

```

GENERATE 1
SAVEVALUE XN1, (MtRand())
TERMINATE 1

PROCEDURE MtRand() BEGIN
TEMPORARY Z;
Z = Call("mtrand.dll", "?nextNum@YAHXZ");
Z = Z/2147483647;
RETURN (Z);
END;

```

Рис. 1. GPSS-программа с вызываемой из нее процедурой MtRand()

```

PROCEDURE MtInit(Arg) BEGIN
Call_Integer("mtrand.dll", "?init@YAHXZ", Arg);
END;

```

Рис. 2. Процедура инициализации генерируемой псевдослучайной последовательности

Uniform(j,0,1), встроенные в GPSS World [1], выдают последовательности, которые начинают самоповторяться, начиная с 2 048 000 000-го числа. При разных j эти ГСЧ выдают одну и ту же (сдвинутую на разные фазы) периодическую последовательность с длиной периода 2 047 999 999. Это не позволяет в экспериментах с большой длиной выборки использовать разные ГСЧ как независимые. На практике, как показывают эксперименты, результаты моделирования (например, оценка среднего времени ожидания заявок в очереди) могут при совместном использовании двух таких ГСЧ существенно смещаться уже при 1 млн обращений к ним. При объемах выборки порядка 1 млн разные ГСЧ начинают «интерферировать», и результаты моделирования существенно искажаются [2]. Поэтому в серьезных имитационных экспериментах на GPSS World (с длинными прогонами модели) использовать более одного ГСЧ в одной модели не приходится, ведь прогон модели с 1 млн обращений к ГСЧ занимает на ПК не более нескольких секунд.

Генераторы класса RNj в GPSS World более разнообразны. Длина периода генераторов RN1 и RN2 составляет 3 145 728 304. Но эти генераторы при совместном использовании тоже могут «интерферировать». Так, начиная с 811 795 903-го числа последовательность RN1 совпадает с началом последовательности RN2. У других ГСЧ класса RNj длина периода может быть другой, но автор не нашел таких ГСЧ, у которых она больше, чем у ГСЧ RN1.

В целом все найденные длины периодов недостаточны для реализации достаточно больших объемов выборок. В статье предлагается способ корректного повышения объемов выборок, доступных в GPSS World, основанный на подключении внешнего датчика. Показано, что это позволяет при моделировании систем массового обслуживания (СМО), описываемых распределениями с тяжелыми хвостами (PTX), значительно расширить возможности решения возникающих здесь проблем [3].

**2. ГСЧ MtRand.** В качестве внешнего генератора псевдослучайных чисел в моделях на GPSS World в настоящее время наиболее целесообразно использовать, по всей видимости, так называемый МТ-генератор, основанный на алгоритме «Вихрь Мерсенна» [4]. В Интернете без особого труда можно найти доступный код этого генератора на языке C++. Из этого кода, для того чтобы генератор

можно было использовать в GPSS World, нужно скомпилировать функцию (файл) \*.dll, поддерживающую протокол CDECL, и поместить ее в папку GPSS World или в папку с выполняемыми моделями \*.gps. Такую функцию можно вызывать из PLUS-процедур [1], используя соответствующий формат команды вызова с указанием имени исполняемого файла и имени функции.

В разработках и исследованиях, описанных ниже, использовался код C++ наиболее распространенной 32-битной версии МТ-генератора, обеспечивающей длину периода  $2^{19\,937} - 1 \approx 10^{6000}$ . Полученная его компиляцией dll-функция, помещенная затем в папку GPSS World, названа mtrand.dll. При вызове этой функции она возвращает целое псевдослучайное число, равномерно распределенное в интервале от 0 до  $(2^{31} - 1)$ . На рис. 1 приведен пример программы с вызовом PLUS-процедуры MtRand(), вызывающей, в свою очередь, функцию mtrand.dll для генерации стандартных случайных чисел.

PLUS-процедура MtRand() реализует здесь стандартную псевдослучайную величину (базовую случайную величину — БСВ), равномерно распределенную в интервале от 0 до 1 с шагом  $\epsilon = 1/(2^{31} - 1)$ . Такой шаг  $\epsilon$  с запасом обеспечивает девять точных десятичных цифр БСВ после запятой. Это заметно повышает точность реализации РТХ (критичную к точности представления БСВ, [5]), так как обычно в GPSS реализуется БСВ только с шестью цифрами после запятой.

В целом программа, представленная на рис. 1, при запуске ее (после трансляции) командой START (n) генерирует n значений БСВ и записывает их в первые n ячеек SAVEVALUE.

Скорость генерации псевдослучайных чисел процедурой MtRand() приблизительно в 2,5 раза меньше, чем скорость стандартной для GPSS World процедуры Uniform. Однако на фоне всех затрат времени на имитационное моделирование это, как правило, приводит к относительно небольшому замедлению. Повысить скорость процедуры MtRand() можно путем применения более быстрых программ, реализующих МТ-генератор. Сообщается, что среди них существуют такие, которые в 2–3 раза превосходят по скорости конгруэнтные генераторы [4], в том числе, следовательно, и генератор Uniform.

Использование MtRand() в каскадных методах реализации РТХ [3, 6–8] также ведет к повышению точности получаемых результатов. Кроме того, сверхбольшой период МТ-генератора делает возможным корректное, практически неограниченное увеличение объема получаемых выборок, что весьма актуально при моделировании СМО с РТХ.

**3. Инициализация ГСЧ MtRand.** Если нужно обеспечить повторную воспроизводимость генерируемой последовательности БСВ, то функцию mtrand.dll можно инициализировать «семенем» (целым положительным числом) с помощью PLUS-процедуры MtInit(s), представленной на рис. 2.

Процедура Mtinit(s) обращается к программе mtrand.dll и передает ей посредством аргумента Arg целое положительное число s для инициализации генератора. Если в разных экспериментах инициализировать генератор одним и тем же числом, то он выдает в них одну и ту же последовательность чисел.

Например, если функция mtrand.dll инициализируется семенем 1, то 10 чисел, формируемых программой (рис. 1) при ее запуске командой START

```

PROCEDURE MtExponential(Arg1,Arg2) BEGIN
TEMPORARY Z, XI;
STEP1: Z = Call("mtrand.dll", "?nextNum@YAHXZ");
IF (Z = 0) THEN GOTO STEP1;
Z = Z/2147483647;
XI = -Arg2 # log(Z) + Arg1;
RETURN (XI);
END;

```

Рис. 3. Процедура реализации экспоненциальной случайной величины

```

PROCEDURE MtPareto(Arg1,Arg2) BEGIN
TEMPORARY Z, XI;
STEP1: Z = Call("mtrand.dll", "?nextNum@YAHXZ");
IF (Z = 0) THEN GOTO STEP1;
Z = Z/2147483647;
XI = Arg1#Z^(-1/Arg2);
RETURN (XI);
END;

```

Рис. 4. Процедура реализации с.в. с распределением Парето

```

GENERATE (MtExponential(0,1))
QUEUE 1
SEIZE 1
DEPART 1
ADVANCE (MtExponential(0,0.9))
RELEASE 1
TERMINATE

GENERATE 1000000
TERMINATE 1

PROCEDURE MtExponential(Arg1,Arg2) BEGIN
TEMPORARY Z, XI;
STEP1: Z = Call("mtrand.dll", "?nextNum@YAHXZ");
IF (Z = 0) THEN GOTO STEP1;
Z = Z/2147483647;
XI = -Arg2 # log(Z) + Arg1;
RETURN (XI);
END;

```

Рис. 5. Модель М/М/1-системы, использующая МТ-генератор

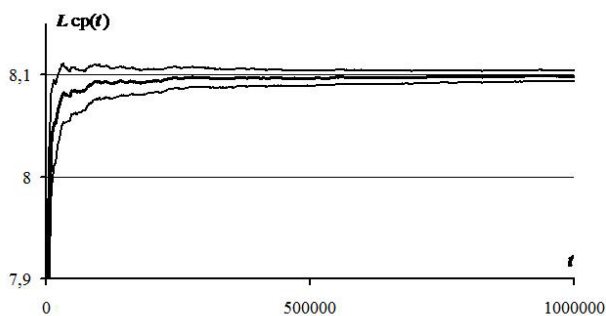


Рис. 6. ПП  $L_{cp}(t)$  (средняя кривая). Верхняя и нижняя кривые ограничивают доверительную полосу, построенную по правилу трех сигм

10, таковы: 0.834044, 0.994370, 0.440649, 0.865115, 0.000229, 0.256249, 0.604665, 0.998081, 0.293512, 0.472178 (здесь числа округлены до 6 цифр после точки).

Если вызвать GPSS World  $N$  раз подряд, то откроется  $N$  окон, и процессы моделирования, запускаемые в этих окнах, будут выполняться на многоядерном компьютере параллельно, по одному на каждом ядре, практически без замедления (при числе ядер, не меньшем  $N$ ). При использова-

нии процедуры MtRand() псевдослучайные последовательности в каждом из окон генерируются своей уникальной копией функции mtrand.dll, инициализируемой (или не инициализируемой) независимо от остальных копий.

Для того чтобы использовать МТ-генератор в GPSS World было более удобно, целесообразно написать библиотеку соответствующих PLUS-процедур, реализующих на основе МТ-генератора разнообразные типовые вероятностные распределения.

**4. Примеры реализации случайных величин с типовыми распределениями.** Возможными значениями на выходе процедуры MtRand() являются как 0, так и 1. Чтобы исключить значение 1, можно в MtRand() величину  $Z$  нормировать делением на 231, а не на  $(2^{31} - 1)$ , как на рис. 1. Для исключения нуля требуется дополнительная проверка, как это делается в представленной на рис. 3 процедуре MtExponential( $s, m$ ), реализующей экспоненциальную случайную величину (с.в.) с математическим ожиданием (м.о.)  $m$  и смещением  $s$ .

Процедура MtPareto( $K, \alpha$ ) реализует с.в. с распределением Парето, где  $K \geq 0$  — наименьшее значение (масштабный параметр),  $\alpha > 0$  — параметр формы. Процедура имеет следующий вид (рис. 4).

В этой процедуре масштабный параметр  $K$  задается аргументом Arg1, параметр формы  $\alpha$  — аргументом Arg2.

Оба приведенных примера генерации с.в. (рис. 3 и рис. 4) используют широко распространенный метод обратного преобразования БСВ (т.е. метод обращения функции распределения [9]).

**5. Пример реализации и обработки выборки большого объема.** Продемонстрировать новые возможности моделирования на GPSS World, приобретаемые благодаря использованию МТ-генератора «Вихрь Мерсенна», можно на простом примере моделирования системы М/М/1 (простейшей СМО). Модель СМО, использующая МТ-генератор посредством обращения к процедуре MtExponential( $s, m$ ) (рис. 3), представлена на рис. 5 (процедура транслируется вместе с использующей ее программой).

На рис. 6 показан график переходного процесса (ПП)  $L_{cp}(t)$  — оценки для средней длины очереди в этой СМО. Значения ПП  $L_{cp}(t)$  получены усреднением 10 000 независимых реализаций процесса изменения в модельном времени  $t$  системного числового атрибута QA1. Моделирование проводилось на интервале от 0 до 1 млн единиц времени. Точки отсчета (значения  $t$ , в которых фиксировались значения QA1) следовали с шагом 1000 единиц времени. Так как в этой простейшей СМО интенсивность поступления заявок  $\lambda$  равна 1, и среднее время их обслуживания  $b$  равно 0,9 (рис. 5), точное значение стационарной средней длины очереди  $L$  составляет  $(\lambda b)^2 / (1 - \lambda b) = 8,1$  [10]. Весь эксперимент на ПК занял около 1,5 часа времени.

Как видно из рис. 6, ПП  $L_{cp}(t)$  сходится к точному стационарному значению  $L = 8,1$ .

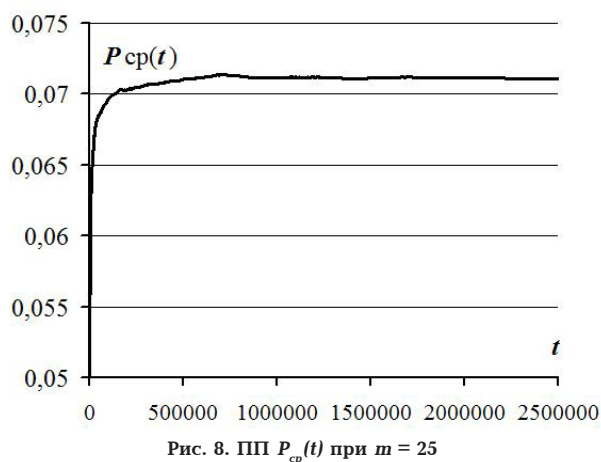
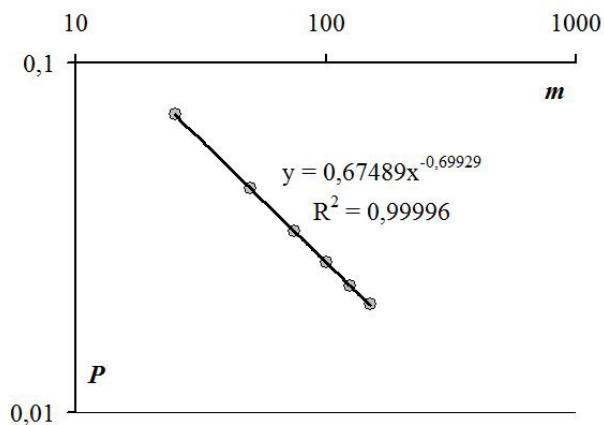
Значение  $L_{cp}(t)$ , достигнутое на момент 1 млн единиц модельного времени, составило 8,0987, близкое к 8,1. Допустимое правилом трех сигм отклонение составило для  $L_{cp}(t)$  на этот момент времени величину  $3 \times 0,00174$ . В пересчете на одну реализацию процесса QA1( $t$ ) это составляет  $3 \times 0,174 \approx 0,52$ . МТ-генератор инициализирован в этом эксперименте семенем 1. Общее число обращений к МТ-генератору составило приблизительно  $2 \times 10^6 \times 10^4 = 20$  млрд, что практически в 10 раз

```

VXOD      GENERATE (MtPareto(1,1.25))
          TEST L Q1,25,ОТКАЗ
          QUEUE 1
          SEIZE 1
          DEPART 1
          ADVANCE (MtPareto(1,1.5))
          RELEASE 1
          TERMINATE
ОТКАЗ    TERMINATE
          GENERATE 1000000
          TERMINATE 1

PROCEDURE MtPareto(Arg1,Arg2) BEGIN
TEMPORARY Z, XI;
STEP1: Z = Call("mtrand.dll", "?nextNum@YAHXZ");
IF (Z = 0) THEN GOTO STEP1;
Z = Z/2147483647;
XI = Arg1#Z^(-1/Arg2);
RETURN (XI);
END;

```

Рис. 7. Модель исследуемой СМО Pa/Pa/1/m при  $m = 25$ Рис. 8. ПП  $P_{cp}(t)$  при  $m = 25$ Рис. 9. Найденная путем ИМ степенная зависимость  $P(m)$ 

больше максимально допустимого числа обращений к стандартному ГСЧ Uniform (равному длине его периода). Таким образом, при использовании Uniform подобное усреднение можно было бы проводить лишь по десятикратному меньшему числу реализаций ПП, получая при этом в  $10^{0,5} \approx 3$  раза большие погрешности (большую ширину доверительной полосы).

Заметим, что использование МТ-генератора, имеющего длину периода около  $10^{6000}$ , в принципе позволяет увеличить число прогонов в этом эксперименте до  $(10^{6000}/20\ 000\ 000\ 000)$  раз, уменьшая за счет этого статистические погрешности получаемых оценок приблизительно в  $10^{3000}$  раз, т.е. обеспечивая около 3 тысяч точных десятичных цифр

после запятой. Конечно, такое увеличение числа прогонов практически невозможно из-за пропорционального ему увеличения времени ИМ. Но и оценки с 3 тысячами точных цифр в ИМ на практике не нужны. Поэтому МТ-генератор при ИМ на GPSS просто обеспечивает практически неограниченные возможности повышения точности за счет увеличения объемов выборки, снимая накладываемые генератором Uniform на длину выборок ограничения, ставшие в настоящее время критическими в связи с быстро растущим быстродействием компьютеров [3, 8].

**6. Пример моделирования СМО с RTX с применением МТ-генератора.** В качестве одной из основных задач моделирования СМО с RTX в [8] обобщается задача определения размера  $m$  буфера, достаточного для того, чтобы стационарная вероятность  $P$  потери заявки не превышала заданной величины. Покажем, как с применением МТ-генератора решается эта задача при ИМ системы Pa/Pa/1/m (где символы Pa соответствуют распределению Парето с порогом  $K$  и параметром формы  $\alpha$ ). Параметры распределений Парето заданы, соответственно, в виде  $K_1 = 1, \alpha_1 = 1,25$  (для входящего потока) и  $K_2 = 1, \alpha_2 = 1,5$  (для времени обслуживания).

Модель этой СМО при размере буфера  $m = 25$  представлена на рис. 7. Процедура MtPareto( $K, \alpha$ ), реализующая RTX Парето и вызываемая из модели, транслируется вместе с использующей ее программой. Выполнен 10001 независимый прогон модели, каждый на интервале модельного времени (0; 2 500 000). При этом произошло около 10 млрд обращений к МТ-генератору «Вихрь Мерсенна». Соответствующая 10001 траектория изменения во времени оценки вероятности отказа (в виде доли потерянных заявок от всех пришедших) накоплена в массиве и перед завершением моделирования усреднена.

Полученная усредненная траектория  $P_{cp}(t)$  изображена на рис. 8. В середине моделируемого интервала времени ПП практически заканчивается. Усредняя во второй половине этого интервала все значения  $P_{cp}(t)$  по времени  $t$ , получаем оценку для вероятности  $P$  при  $m = 25$ , составляющую 0,07114.

Аналогичные эксперименты при  $m = 50, 75, 100, 125$  и  $159$  дают для  $P$  оценки 0,0438, 0,0329, 0,0269, 0,0230 и 0,0204 соответственно. На получение этих шести точек ( $m, P$ ) зависимости  $P(m)$  потрачено около суток компьютерного времени. Все шесть точек идеально легли в координатной плоскости с логарифмическими масштабами осей на прямую (рис. 9). Это подтверждает высокую точность полученных оценок для  $P(m)$ , поскольку эта зависимость при больших  $m$  является степенной [11]. Получаемая с высокой точностью линия тренда дает для  $P(m)$  уравнение  $P = 0,675m - 0,7$ . Отсюда легко выводится искомая зависимость  $m(P)$ :  $m = (P/0,675) - (1/0,7) \approx 0,57P - 1,43$ , которую можно использовать для расчета требуемых  $m$  при любых  $P \leq 0,0438$ .

Учитывая, сколь большое число обращений к ГСЧ потребовалось в описанном эксперименте, нетрудно заметить, что если бы в нем использовались стандартные ГСЧ, встроенные в GPSS World, то такая точность при моделировании не могла бы быть достигнута. Ознакомление с прочими преимуществами МТ-генератора только укрепляет этот вывод [4]. На практике он подтверждается и большим количеством неудач, постигших автора статьи при использовании стандартных ГСЧ GPSS World



при ИМ СМО с РТХ. Ни в одном таком опыте статистические оценки зависимостей  $P(m)$  не ложились столь очевидным образом на прямую, как это происходит в опытах с применением генератора MtRand (рис. 9).

**7. Выводы.** Ограниченные возможности генераторов случайных чисел языка GPSS World не позволяют в полной мере реализовать быстродействие современных компьютеров для повышения точности результатов имитационного моделирования. Эту проблему можно решить применением внешних генераторов.

Хорошим выходом является применение МТ-генератора с длиной периода ( $2^{19} - 1$ ), основанного на алгоритме «Вихрь Мерсенна».

Изложенные в статье подходы и опыт применения МТ-генератора показывают, что этот генератор позволяет качественно повысить точность и надежность результатов моделирования СМО на GPSS World, в том числе таких СМО, которые описываются распределениями с тяжелыми хвостами.

#### Библиографический список

1. Руководство пользователя по GPSS World / Под ред. К. В. Кудашова ; пер. с англ. — Казань : Мастер Лайн, 2002. — 384 с.
2. Задорожный, В. Н. Реализация больших выборок при моделировании систем массового обслуживания на GPSS World / В. Н. Задорожный // ИММОД-2015 : материалы тр. VII Всерос. науч.-практ. конф. В 2 т. Т. 1. — М., 2015. — С. 225—229.
3. Задорожный, В. Н. Особенности моделирования систем массового обслуживания с тяжелыми хвостами распределений на GPSS World. Метод ARAND / В. Н. Задорожный // Омский научный вестник. Сер. Приборы, машины и технологии. — 2015. — № 3 (143). — С. 307—311.

4. Вихрь Мерсенна [Электронный ресурс]. — Режим доступа : <http://dic.academic.ru/dic.nsf/ruwiki/88739> (дата обращения: 20.12.2015).

5. Задорожный, В. Н. Проблемы генерации случайных величин с фрактальными распределениями / В. Н. Задорожный, О. И. Кутузов // Омский научный вестник. Сер. Приборы, машины и технологии. — 2012. — № 3 (113) — С. 20–24.

6. Zadorozhnyi, V. N. Cascade Method of Realization of Heavy-Tailed Distributions in Data Network Modelling. 2015 International Siberian conference on control and communications SIBCON, sec. Control of the Large-Scale Systems, Russia, Omsk, May 21—23, 2015.

7. Задорожный, В. Н. Каскадный метод реализации распределений с тяжелыми хвостами / В. Н. Задорожный // Омский научный вестник Сер. Приборы, машины и технологии. — 2015. — № 140. — С. 222—226.

8. Zadorozhnyi V. N. Fractal Queues Simulation Peculiarities / Communication in Computer and Information Science, Vol. 564 : Information Technologies and Mathematical Modelling, 14th International Scientific Conference, ITMM 2015.

9. Харин, Ю. С. Практикум на ЭВМ по математической статистике / Ю. С. Харин, М. Д. Степанова. — Мн., 1987. — 304 с.

10. Клейнрок, Л. Вычислительные системы с очередями / Л. Клейнрок ; пер. с англ. Б. С. Цыбакова. — М. : Мир, — 1979. — 600 с.

11. Zadorozhnyi V. N. Simulation modeling of fractal queues / Dynamics of Systems, Mechanisms and Machines (Dynamics), 2014 DOI: 10.1109/Dynamics.2014.7005703 Publication Year: 2014, P. 1—4.

**ЗАДОРЖНЫЙ Владимир Николаевич**, доктор технических наук, доцент (Россия), профессор кафедры «Автоматизированные системы обработки информации и управления».

Адрес для переписки: [zwn2015@yandex.ru](mailto:zwn2015@yandex.ru)

Статья поступила в редакцию 21.12.2015 г.

© В. Н. Задорожный

## Книжная полка

621.39/B27

**Величко, В. В. Модели и методы повышения живучести современных систем связи / В. В. Величко, Г. В. Попков, В. К. Попков. — М. : Горячая линия-Телеком, 2014. — 269 с. — ISBN 978-5-9912-0408-8.**

Рассмотрены вопросы анализа живучести сетей связи в условиях разрушающих информационных воздействий. Дана классификация информационных атак в информационных сетях и методы их обнаружения. Уделено значительное внимание вопросам, связанным с живучестью и надёжностью мобильных систем связи, предложены модели структурной надёжности.

004.4/B14

**Вайсфельд, М. Объектно-ориентированное мышление / М. Вайсфельд ; пер. с англ. В. Черник. — СПб. : Питер, 2014. — 303 с. — ISBN 978-5-496-00793-1.**

Объектно-ориентированное программирование — это фундамент современных языков программирования, включая C++, Java, C#, Visual Basic, .NET, Ruby и Objective-C. Кроме того, объекты лежат в основе многих веб-технологий, например JavaScript, Python и PHP. Объектно-ориентированное программирование обеспечивает правильные методики проектирования, переносимость кода и его повторное использование, однако для того, чтобы все это полностью понять, необходимо изменить свое мышление. Разработчики, являющиеся новичками в сфере объектно-ориентированного программирования, не должны поддаваться искушению перейти непосредственно к конкретному языку программирования (например, Objective-C, VB .NET, C++, C#, .NET или Java) или моделирования (например, UML), а вместо этого сначала уделить время освоению того, что автор книги Мэтт Вайсфельд называет объектно-ориентированным мышлением. Несмотря на то, что технологии программирования изменяются и эволюционируют с годами, объектно-ориентированные концепции остаются прежними — при этом неважно, какой именно является платформа.

## ПЕРЕХОДНЫЕ ПРОЦЕССЫ В РАСТУЩИХ СЕТЯХ С НЕЛИНЕЙНЫМ ПРАВИЛОМ ПРЕДПОЧТИТЕЛЬНОГО СВЯЗЫВАНИЯ

Выполняется анализ переходных процессов в растущих стохастических сетях с нелинейным правилом предпочтительного связывания. Выводятся уравнения динамики развития сетей. Разрабатываются численные методы расчета переходных процессов в сетях и выделенных узлах сетей.

**Ключевые слова:** растущие сети, случайные графы, стационарные и переходные случайные процессы.

**1. Введение.** В работах [1–3] показано, что для больших сетей (таких как WWW, сеть сотрудничества ученых, сеть передачи электроэнергии и др.) классический случайный граф Эрдеша–Реньи [4] не является адекватной моделью. Поэтому в современной теории сетей (*Network Science*) [5, 6] разрабатываются новые математические модели — новые виды случайных графов, описывающих и объясняющих свойства реальных больших сетей, такие как их малый диаметр, степенное распределение степеней связности узлов и т.д. Многие структурные свойства реальных больших сетей удалось объяснить их развитием по простому общему правилу — правилу предпочтительного связывания [7]. Согласно этому правилу сеть растет в результате добавления новых узлов, которые связываются с имеющимися узлами, предпочитая те, у которых степень связности  $k$  выше (правило «богатый становится богаче» [8]).

В настоящее время при исследовании больших сетей широко распространена модель, называемая случайным графом Барабаши–Альберт (граф БА) [7, 8]. Граф БА выращивается из произвольного небольшого связного графа (затравки) путем добавления к нему в моменты  $t = t_1, t_2, \dots$  приращения графа — очередной новой вершины с  $m$  инцидентными ей ребрами.

Свободные концы ребер добавляемого приращения связываются со случайно выбираемыми вершинами графа. В графе БА вероятность  $p_i$  того, что ребро выберет для связи вершину  $i$ , пропорциональна ее локальной степени связности  $k_i$ :

$$p_i = \frac{k_i}{\sum_j k_j}. \quad (1)$$

Бесконечное добавление приращений приводит к выращиванию бесконечного графа БА, который называется безмасштабным, потому что распределение степеней связности (РСС) его вершин является асимптотически степенным, т.е. безмасштабным распределением. Такое РСС вершин графа БА согласуется с РСС узлов многих, хотя и далеко не всех реальных сетей [9, 10].

Тем не менее подход, предложенный А. Барабаши и Р. Альберт, позволяет улучшить согласование РСС графов и РСС моделируемых сетей путем перехода от линейного правила (1), в котором  $p_i \propto k_i$ , к нелинейному правилу предпочтительного связывания (НППС), в котором  $p_i \propto f(k_i)$ :

$$p_i = \frac{f(k_i)}{\sum_j f(k_j)}, \quad (2)$$

где  $f(k) \equiv f_k$  — произвольная неотрицательная весовая функция степени  $k$  вершины. В начале 2000-х годов изучались графы, выращиваемые при использовании конкретных классов нелинейной функции  $f(k)$ . Выдающиеся результаты в этом направлении были получены Л. Крапивским и С. Реднером [11], которые, применяя уравнения статистической механики, описали и всесторонне исследовали графы с НППС при  $m = 1$  и  $f(k) = k^\gamma$ ,  $0 < \gamma < 1$ .

Основы общей теории графов с НППС, не накладывающей ограничения на вид весовой функции  $f(k) \equiv f_k$ , были разработаны в статье [12]. Здесь обобщен также вид возможных приращений графа и всесторонне исследованы графы с НППС, выращиваемые добавлением стохастических приращений со случайным целым числом  $x \geq 1$  ребер, имеющим произвольное распределение вероятностей  $r_k = P\{x = k\}$ . Решена задача анализа моделей — определения РСС графа, выращиваемого при заданных  $f(k)$  и  $\mathbf{r} = \{r_k\}$ , и задача синтеза моделей — нахождения  $f(k)$  и  $\mathbf{r}$ , обеспечивающих заданное РСС выращиваемого графа (например, такое же, как в моделируемой большой сети). В [13] рассмотрен пример применения теории графов с НППС к исследованию транспортных сетей. В [14] методы и результаты теории графов с НППС существенно расширены и систематизированы, в частности, решена задача определения совместного РСС пары вершин, связанных случайно выбранным ребром.

Актуальность теории графов с НППС обусловлена широкими возможностями ее практического применения для решения глобальных задач современности, включая задачи инфокоммуникационных технологий, энергетики, транспорта, а также задачи

мониторинга и регулирования социальных сетей, в том числе задачи информационного противоборства.

Для успешного решения этих задач необходимо не только уметь правильно рассчитывать стационарные характеристики уже сформированной большой сети, но и выполнять анализ переходных процессов в сети (ПП), характеризующих динамику начальных стадий ее развития. Такой анализ позволит предсказывать темпы развития сети и определять своевременные оптимальные воздействия, обеспечивающие приемлемую траекторию ее развития и гарантирующие достижение требуемого управленческого эффекта при надежном решении проблем безопасности.

В общем случае анализ ПП в сетях с НППС можно выполнять методами имитационного моделирования (ИМ). Однако высокая цена точности ИМ больших сетей, определяемая вычислительными затратами (часто непомерными), существенно ограничивает не только возможности его практического применения, но и самые постановки решаемых задач (например, если речь идет о задачах управления ростом сетей). В данной статье разрабатываются численные методы анализа ПП в графах с НППС, радикально повышающие скорость и точность выполняемого анализа.

**2. Задача расчета сетевых переходных процессов.** Распределением степеней связности (РСС) вершин графа будем называть ряд (последовательность)  $\mathbf{q} = \{q_k\} = \{q_1, q_2, \dots\}$  вероятностей  $q_k$  того, что случайно (равновероятно) выбранная вершина графа имеет степень связности  $k$ . Например, граф из трех вершин, связанных двумя ребрами, имеет РСС  $\mathbf{q} = \{q_1, q_2\} = \{2/3, 1/3\}$ , а полный граф из 4 вершин — РСС  $\mathbf{q} = \{0, 0, 0, 1\}$ . Графы с изолированными вершинами мы не рассматриваем.

Сетевым ПП будем называть изменение РСС во времени:  $\mathbf{q}(t) = \{q_k(t)\}$ ,  $t = 0, 1, 2, \dots$ , начинающееся с некоторого момента  $t = t_0$ . Далее будем считать, что  $t_0 = N_0$ , где  $N_0$  — число вершин в затравке. Тогда в ходе ПП число  $N$  вершин в растущем графе всегда равно текущему времени:  $N(t) = t$ , и вершины, добавляемые к графу, можно нумеровать моментами их добавления.

Итак, пусть в момент времени  $t$  граф содержит  $N(t) = t$  вершин и имеет РСС  $\mathbf{q}(t)$ . В момент  $t + 1$  поступает новая вершина с  $x \in \{r_k\}$  ребрами, которые связываются с имеющимися вершинами графа, выбирая их случайно по правилу (2). В результате РСС  $\mathbf{q}(t)$  преобразуется в РСС  $\mathbf{q}(t + 1)$ .

Задачу расчета сетевых ПП в графах с НППС поставим как задачу определения  $\mathbf{q}(t + 1)$  при заданных  $f(k)$ ,  $\{r_k\}$  и  $\mathbf{q}(t)$ .

**3. Решение задачи расчета сетевых ПП.** Задачу расчета сетевых ПП можно решить путем незначительной модификации метода рассуждений, разработанного в [12] для вывода финальных (стационарных) РСС  $\mathbf{Q} = \lim_{t \rightarrow \infty} \mathbf{q}(t)$  при  $t \rightarrow \infty$ .

Следуя работе [12], обозначим через  $A_k$  множество вершин графа (слой), имеющих степень  $k$ ,  $k \geq g$  (где  $g \geq 1$  — минимальный номер слоя), согласуя введение параметра  $g$  с условиями равенства нулю вероятностей  $r_k$  и весов  $f(k)$  при  $k < g$ , а также с отсутствием в затравке вершин со степенью, меньшей  $g$ .

В соответствии с НППС (2) вероятность  $P_k(t)$  того, что при добавлении приращения к графу с  $N = t$  вершинами новое ребро выберет вершину слоя  $A_k$ , составляет

$$P_k(t) = \sum_{i \in A_k} p_i = \frac{\sum_{i \in A_k} f(k_i)}{\sum_{j=1}^N f(k_j)} = \frac{f(k)|A_k|}{\sum_{l \geq g} f(l)|A_l|} = \frac{f_k |A_k| / N}{\sum_{l \geq g} f_l |A_l| / N} \sim \frac{q_k(t) f_k}{\bar{f}(t)} = \frac{q_k(t) f_k}{\bar{f}(t)}, \quad (3)$$

где  $|A_k|$  — число вершин в слое  $A_k$ ,  $\bar{f}(t)$  — средний вес вершин графа в момент  $t$ .

При добавлении к графу с  $N$  вершинами приращения, приходящего в момент  $t + 1$  и содержащего в среднем  $M(x) = m$  ребер, происходят следующие изменения:

— новая вершина приращения попадает с вероятностью  $r_k$  в слой  $A_k$ , в результате чего среднее число  $N_k$  вершин в нем возрастает на  $r_k$  (т.е. с вероятностью  $r_k$  возрастает на 1 и с вероятностью 0 возрастает на 0),  $k \geq g$ ;

— вместе с тем каждый слой  $A_k$  теряет в среднем  $mP_k(t)$  вершин, переходящих в следующий слой вследствие того, что какие-нибудь из  $x$  новых ребер присоединяются к вершинам этого слоя.

Действительно, первое новое ребро связывается с вершиной слоя  $A_k$  с вероятностью  $P_k(t)$ , степень вершины повышается, она уходит в следующий слой  $A_{k+1}$ , и, следовательно, за счет одного нового ребра слой  $A_k$  теряет в среднем  $P_k(t)$  вершин. А за счет всех  $x$  ребер приращения — в среднем  $M[xP_k(t)] = mP_k(t)$  вершин.

Считая  $N$  достаточно большим, мы пренебрегаем здесь вероятностью того, что одна вершина может быть выбрана более чем одним ребром приращения<sup>1</sup>, а также изменением вероятности  $P_k(t)$  до завершения привязывания всех ребер приращения;

— кроме того, каждый слой  $A_k$ , исключая слой  $A_g$ , приобретает в среднем  $mP_{k-1}(t)$  новых вершин, приходящих из предыдущего слоя  $A_{k-1}$ .

Учитывая все эти изменения среднего числа  $N_k$  вершин в слоях, получаем:

$$\begin{aligned} N_g(t + 1) &= N_g(t) + r_g - mP_g(t), \quad k = g, \\ N_k(t + 1) &= N_k(t) + r_k + mP_{k-1}(t) - mP_k(t), \\ &k > g. \end{aligned} \quad (4)$$

Подставляя  $N_k(t) = N(t)q_k(t) = tq_k(t)$  и  $N_k(t + 1) = N(t + 1)q_k(t + 1) = (t + 1)q_k(t + 1)$  (при всех  $k \geq g$ ) и деля оба полученных равенства на  $(t + 1)$ , находим:

$$\begin{aligned} q_g(t + 1) &= \frac{tq_g(t) + r_g - mP_g(t)}{t + 1}, \\ q_k(t + 1) &= \frac{tq_k(t) + r_k + mP_{k-1}(t) - mP_k(t)}{t + 1}, \quad k > g. \end{aligned}$$

Наконец, используя здесь последнее в (3) выражение для  $P_k(t)$ , получаем искомое решение задачи расчета сетевых ПП:

$$q_g(t + 1) = \frac{tq_g(t) + r_g - m \frac{q_g(t) f_g}{\bar{f}(t)}}{t + 1},$$

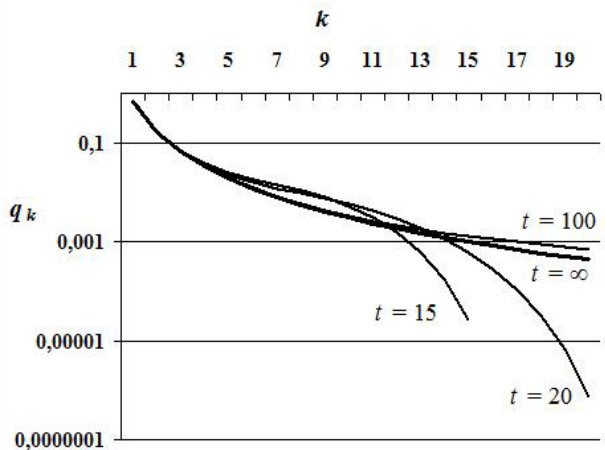


Рис. 1. Изменение во времени РСС  $q = \{q_k\}$  графа БА при  $m = 1$  (дерева БА)

$$q_k(t+1) = \frac{tq_k(t) + r_k + \frac{m}{f(t)} [q_{k-1}(t)f_{k-1} - q_k(t)f_k]}{t+1}, \quad k > g, \quad (5)$$

где  $\bar{f}(t) = \sum_{k \geq g} q_k(t) f_k$ .

**4. Примеры расчета сетевых ПП.** Соотношения (5) позволяют легко рассчитывать РСС вершин графов с НППС для времени  $t > t_0$  начиная с РСС, известного для  $t = t_0$ . Их удобно использовать для расчета сетевых ПП в Excel, поскольку строка РСС  $q(t+1)$  рассчитывается непосредственно через предыдущую строку  $q(t)$ . Не зависящие от времени ряды значений  $\{k\}$ ,  $\{r_k\}$  и  $\{f_k\}$  фиксируются в строках «шапки» таблицы. Скалярные показатели, определяемые через РСС  $q(t)$ , в частности, средняя степень вершин  $\bar{k}(t) = \sum k q_k(t)$  и их средний вес  $\bar{f}(t) = \sum_{k \geq g} f_k q_k(t)$ , вычисляются в столбцах, выделяемых в левой части таблицы.

Правильность соотношений (5), выведенных для расчета сетевых ПП, подтверждается результатами ИМ различных графов с НППС. Кроме того, при расчете в Excel сетевых ПП в различных графах с НППС,  $q(t)$  всегда очевидным образом сходится к финальному РСС  $Q$ , которое рассчитывается по формулам, найденным в работах [12–14] (а для случая  $f_k = k^g$  — и в работе [11]).

На рис. 1 показаны графики РСС  $q(t)$ , рассчитанного по формулам (5) при  $f(k) = k$  (линейное правило предпочтительного связывания) и  $r_i = 1$  (т.е. при фиксированном приращении с числом ребер  $x \equiv m = g = 1$ ). Для наглядности дискретные ряды  $\{q_k\}$  ( $k = 1, \dots, N$ ) представлены непрерывными линиями.

На рис. 2 показаны графики РСС  $q(t)$ , рассчитанного по формулам (5) при использовании весовой функции  $f(k)$ , равной  $k \ln(k)$  при  $2 \leq k \leq 20$  и равной нулю при прочих  $k$ . При этом для выращивания графа используются стохастические приращения с распределением вероятностей числа ребер  $r_2 = 0,1, r_3 = 0,2, r_4 = 0,3, r_5 = 0,4$ , а заправка состоит из 5 вершин, связанных 5 ребрами в кольцо.

На шаге  $t = 10\,000$  (черные столбики на рис. 2) РСС совпадает с точностью до 4 значащих цифр

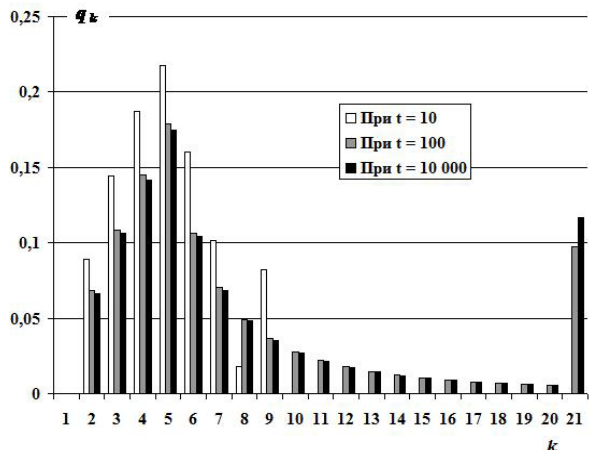


Рис. 2. Динамика РСС графа с весовой функцией  $f(k) = k \ln(k), 2 \leq k \leq 20$  ( $f(k) = 0$  для других  $k$ ), при  $r = \{r_1, r_2, r_3, r_4, r_5\} = \{0; 0,1; 0,2; 0,3; 0,4\}$

с финальным РСС, рассчитанным по формулам статьи [12].

**5. Расчет локальных переходных процессов.** Выделим произвольную вершину, присоединенную к растущему графу. Момент присоединения выделенной вершины (ВВ) назовем временем ее рождения и обозначим через  $t_{ВВ}$ . Время  $t_{ВВ}$  одновременно является номером ВВ. С возрастом  $T = t - t_{ВВ}$  состояние (степень  $k$ ) ВВ случайным образом возрастает. Локальным ПП будем называть изменение распределения вероятностей состояний ВВ (РСВВ)  $u(T) = \{u_k(T)\} = \{u_1(T), u_2(T), \dots\}$  с ее возрастом  $T$ .

Пусть в момент  $T$  времени жизни ВВ ее степень  $k = k(T)$ . Тогда добавляемая к графу в момент  $T+1$  новая вершина при условии, что она имеет  $x$  инцидентных ей ребер, присоединится к ВВ одним из них с вероятностью

$$p_{ВВ|x} \sim \frac{x f(k)}{\sum_j f(k_j)} = \frac{x f(k)}{N \bar{f}(t)} = \frac{x f(k)}{t \bar{f}(t)}. \quad (6)$$

Безусловная по  $x$  вероятность увеличения на 1 степени  $k$  ВВ равна математическому ожиданию условной вероятности (6) и составляет

$$p_{ВВ} = p_{k,k+1} \sim \frac{m f(k)}{t \bar{f}(t)}, \quad (7)$$

где  $t = T + t_{ВВ}$  — текущее время.

$\bar{f}(t) = \sum_k f(k) q_k(t)$  — средний вес вершин графа в момент времени  $t$ , определяемый при расчете сетевого ПП графа.

Вероятность (7) перехода ВВ на шаге  $T+1$  из состояния  $k$  в состояние  $k+1$  является основной переходной вероятностью, определяющей неоднородную марковскую цепь с дискретным временем. По начальному РСВВ  $u(0)$  можно вычислять РСВВ в последующие моменты времени  $T$ , используя формулу полной вероятности. С учетом того, что ВВ на каждом шаге времени может либо сохранить текущее состояние (степень) либо увеличить на 1, вероятности  $u_k(T+1)$  вычисляются через вероятности  $u_k(T)$  следующим образом:

$$u_g(T+1) = u_g(T) [1 - p_{g,g+1}], \quad k = g,$$

$$u_k(T+1) = u_{k-1}(T) p_{k-1,k} + u_k(T) [1 - p_{k,k+1}], \quad k > g.$$



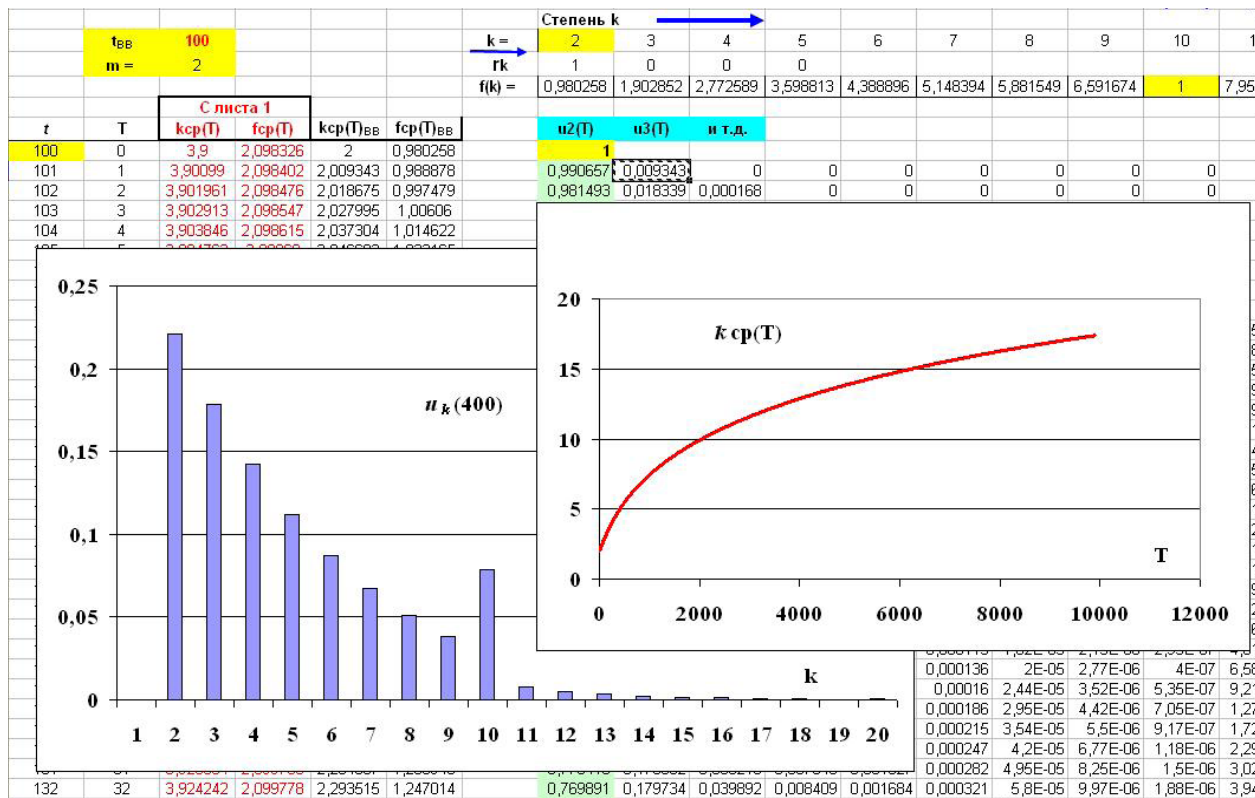


Рис. 3. Лист Excel с результатами расчета РСВВ  $u(400)$  и ПП  $k_{cp}(T)$  (локальный ПП  $k_{cp}(T)$  при  $f(k) = k$  изучен в [15] аналитическими методами)

С учетом (7) отсюда находим:

$$u_g(T+1) = u_g(T) \left( 1 - \frac{mf(g)}{(T+t_{BB})\bar{f}(T+t_{BB})} \right), \quad k = g. \quad (8)$$

$$u_k(T+1) = u_{k-1}(T) \frac{mf(k-1)}{(T+t_{BB})\bar{f}(T+t_{BB})} + u_k(T) \left( 1 - \frac{mf(k)}{(T+t_{BB})\bar{f}(T+t_{BB})} \right), \quad k > g.$$

Расчет локального ПП ВВ по формулам (8) легко реализуется в Excel.

**6. Примеры расчета локальных ПП.** На рис. 3 показано распределение  $u(T) = u(400)$  вероятностей состояний ВВ, «родившейся» в момент времени  $t = t_{BB} = 100$  в графе с НППС, выращиваемом при фиксированных приращениях с числом ребер  $m = 2$  и с весовой функцией

$$f(k) = \begin{cases} k^{\frac{1}{2}} \ln k & \text{при } k \geq 2, \text{ если } k \text{ не кратно } 10, \\ 1 & \text{при } k \geq 2, \text{ если } k \text{ кратно } 10, \\ 0 & \text{при } k < 2 \end{cases}$$

из затравки, содержащей 5 вершин, соединенных в кольцо.

Здесь же показан ПП изменения средней степени связности ВВ  $k_{cp}(T)$ .

Расчет подтверждается результатами ИМ. Выполненный предложенным в статье методом расчет выполняется компьютером за малую долю секунды. Расчет методом ИМ (с приемлемой погрешностью оценок вероятностей  $q_k$  порядка 10–20 %) требует выполнения 5000 прогонов модели и занимает около 1,5 часа. Таким образом, разработанный метод

позволяет по сравнению с ИМ при несоизмеримом повышении точности результатов сократить затраты времени более чем на пять порядков. Затраты компьютерной памяти при использовании разработанного метода растут как  $O(N)$ , а при ИМ — как  $O(N^2)$ .

**7. Заключение.** Полученные в статье уравнения динамики (5) и (8) позволяют эффективно выполнять численными методами анализ сетевых и локальных переходных процессов в графах с НППС, являющихся обобщением графов БА и охватывающих в качестве частных случаев другие обобщения специального вида, в том числе графы с фиксированным приращением и степенной весовой функцией [11].

Достигнутые быстрота и точность расчета переходных процессов позволяют в рамках теории сетей (*Network Science*) ставить и решать задачи анализа и синтеза широкого класса больших стохастических сетей, ранее недоступные для решения точными математическими методами. К таким задачам относятся, в частности, задачи динамической идентификации растущих сетей (включая социальные сети, транспортные, торговые, террористические сети и т.д.) и задачи выбора и оптимизации управленческих решений, актуальные в отношении широкого класса таких сетей.

#### Примечание

<sup>1</sup> Соотношение (3) не исключает существования таких весовых функций  $f(k)$ , которые обеспечивают лидирующей вершине (первую оказавшейся в слое  $A_k$  с максимальным из непустых слоев номером  $k$ ) вероятность последующего выбора ее для связывания, сколь угодно близкую к единице. Такая вершина (назовем ее «кадавром» [16]) будет присоединять к себе все или почти все  $x$  ребер приращения и перемещаться

за один шаг из слоя  $A_k$  сразу в слой  $A_{k+x}$ , в результате чего на следующем шаге вероятность выбора ее для связывания станет еще более высокой. В таких случаях выращивается большой граф, все или почти все вершины которого оказываются связанными с «кадавром». В статьях [11, 14] показано, что к таким сингулярным (особым) случаям относится, например, случай весовой функции  $f(k) = k^2$ . Сингулярные случаи выращивания графов мы в данной статье не рассматриваем.

#### Библиографический список

1. Watts D. J., Strogatz S. H. Collective dynamics of small-world networks, *Nature* 393, 440–442 (1998).
2. Albert R., Jeong H., Barabasi A. L. Diameter of the world-wide web, *Nature* 401, 130–131 (1998).
3. Newman M., The structure of scientific collaboration networks, *Proc. Natl. Acad. Sci. USA* 98, 404–409 (2001).
4. Erdos P., Renyi A. On the evolution of random graphs, *Publications of the Mathematical Institute of the Hungarian Academy of Sciences* 5, 17–61 (1960).
5. Barabasi A. L. Scale-free networks: A decade and beyond, *Science* 325, 412–413 (2009).
6. Barabasi A. L. Network science, *Philosophical Transactions of The Royal Society* 731, 1–3 (2013).
7. Barabasi A. L., Albert R. Emergence of scaling in random networks, *Science* 286, 509–512 (1999).
8. Barabasi A. L., Albert R. Statistical mechanics of complex networks, *Rev. Mod. Phys* 74, 47–97 (2002).
9. Clauset A., Shalizi C. R., Newman M. Power-law distributions in empirical data, *Rev. Mod. Phys* 51, 661–703 (2009).
10. Amaral L. A. N., Scala A., Barthélemy M., Stanley H. E., *Classes of small-world networks*, in: *Proc. Natl. Acad. Sci. U.S.A.* 97, 2000, p. 11149.
11. Krapivsky P. L., Redner S. Organization of growing random networks, *Phys. Rev. E* 63, 066123 (2001).
12. Задорожный, В. Н. Случайные графы с нелинейным правилом предпочтительного связывания / В. Н. Задорожный / *Проблемы управления*. – 2011. – № 6. – С. 2–11.
13. Zadorozhnyi V., Yudin E. Growing Network: Nonlinear Extension of the Barabasi-Albert Model // *Communications in Computer and Information Science*, 2014. – Т. 487. – С. 432–439.
14. Zadorozhnyi V. N., Yudin E. B. Growing network: models following nonlinear preferential attachment rule, *Physica A: Statistical Mechanics and its Applications*, v. 428, pp. 111–132, 2015. – DOI: 10.1016/j.physa.2015.01.052.
15. Задорожный, В. Н. Исследование динамики роста степени связности вершин случайного графа в моделях виртуальных сетей / В. Н. Задорожный, В. А. Бадрызов // *Омский научный вестник. Сер. Приборы, машины и технологии*. – 2015. – № 1 (137). – С. 215–219.
16. Стругацкие, А. Н. и Б. Н. Понедельник начинается в субботу. Повесть-сказка для научных работников младшего возраста / А. Н. и Б. Н. Стругацкие. – М.: ДЛ, 1965. – 135 с.

**ЗАДОРЖНЫЙ Владимир Николаевич**, доктор технических наук, доцент (Россия), профессор кафедры «Автоматизированные системы обработки информации и управления».  
Адрес для переписки: zwn2015@yandex.ru  
**БАДРЫЗЛОВ Владимир Александрович**, аспирант кафедры «Автоматизированные системы обработки информации и управления».  
Адрес для переписки: v\_bad@mail.ru

Статья поступила в редакцию 18.12.15  
© В. Н. Задорожный, В. А. Бадрызов

УДК 167/168.0001.8+510:514.8:515.1:519.1/2/6/7+ В. П. СИЗИКОВ  
+53+550.36+577.31

Омский государственный  
университет путей сообщения

## СИСТЕМНЫЙ АНАЛИЗ ВЗАИМОДЕЙСТВИЯ ДВУХ ТЕЛ НА БАЗЕ ДИС-ТЕХНОЛОГИИ

С привлечением системной методологии в ранге ДИС-технологии осуществлен системный анализ взаимодействия двух тел. На примере модели в ранге триады выявлены особенности в проявлении такого взаимодействия.

**Ключевые слова:** гравитация, ДИС-технология, процесс-система, режим, электромагнетизм.

**1. Введение.** Данная работа является продолжением исследований [1–6]. Ключевыми моментами здесь выступают определенность с математическим эквивалентом понятия системы [2] и представление о процессе-системе как круговороте ресурса [5–6]. Именно эти моменты сыграли решающую роль в разработке на базе теории динамических информационных систем (ДИС, ТДИС) [7] инструментария ДИС-технологии, которому присущи одновременно качества и системной методологии, и языка программирования на уровне оболочки экспертных

систем, и аппарата имитационного моделирования [3–5].

Одним из главных недостатков современной науки, включая физику, является отсутствие работ на системном уровне. Продолжаются тенденции полного сведения сущности системы к её проявлениям на внешнем уровне. Внутренним атрибутам системы внимание практически не уделяется, её продолжают считать чёрным ящиком. Потому не системны и традиционные представления о движении. В них принято равномерное прямолинейное

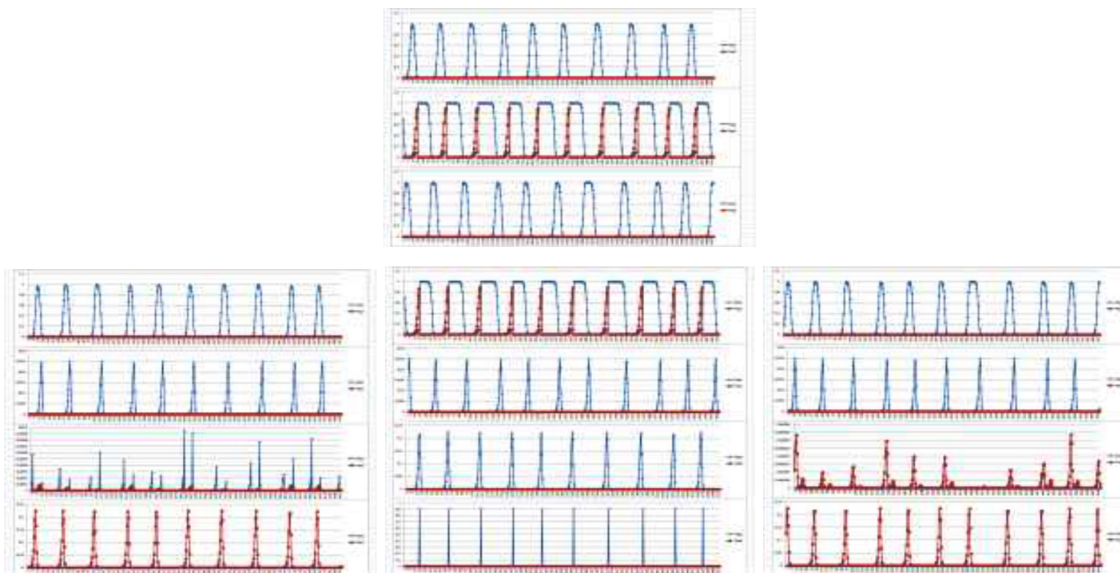


Рис. 1. Базовый режим взаимодействия основного тела и спутника

и вращательное движение объединять в одно — механическое движение. Но на внутреннем уровне системы эти два движения проистекают из разных миров — первое из актива, а второе — из пассива [5], т.е. единить их вместе нет смысла. На деле механическое движение есть посредник, связующий указанные два типа. И, конечно, движение на внутреннем уровне должно объединять аспекты кинетической и потенциальной энергии, т.е. без учёта напряжений, ёмкостей здесь обходиться тоже не имеет смысла. Как-никак, но не система теперь движется относительно чего-то, а в самой системе происходят движения.

Инструментарий ДИС-технологии позволяет быстро получать на качественном уровне то, что традиционно требует долгих лет исследований, проведения чрезвычайно затратных и экологически вредных экспериментов. В данной работе предлагается продолжить апробацию этого инструментария [8–12] на примере задачи взаимодействия двух тел, начала чему подавались в [1].

#### Модель взаимодействия тела и его спутника.

Рабочим объектом ДИС-технологии является ДИС как оргграф с двумя типами рёбер (ведущими, контролирующими) и процессом информационного функционирования (ПИФ) на нём как процессом перераспределения между вершинами двух типов ресурса (актива, пассива) в последовательности компонентов из актов трёх типов:  $A_c$  — акт сбора актива в пассив по контролирующим рёбрам ДИС;  $A_t$  — акт трансформации пассива в актив в некоторых вершинах ДИС;  $A_d$  — акт перераспределения актива по ведущим рёбрам ДИС.

Ограничимся моделью в форме триады с вершинами  $v_0, v_1, v_2$ , где  $v_0$  представит спутник,  $v_1$  — основное тело,  $v_2$  — среду. Ресурс в триаде свершает кругооборот по циклу  $v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow v_0$ . Для запуска ПИФ надо задать начальное распределение активов  $r_i$  и пассивов  $q_i$  по вершинам  $v_i$  ( $i=0,1,2$ ) и то, как меняются от компонента к компоненту ПИФ уровни трансформации  $\lambda_i$  в каждой вершине  $v_i$  и относительные проводимости  $f_c(v_j, v_i), f_d(v_i, v_j)$  контролирующего и ведущего ребра, проводящего ресурс от  $v_i$  к  $v_j$ . Для здорового анализа считаем, что общий объём ресурса в триаде равен 1 — условной

единице. Кроме того, режим, на который выходит ПИФ триады, почти всегда не зависит от начальных значений  $r_i$  и  $q_i$ , т.е. их можно задать практически любыми, дающими в сумме 1.

Далее, ввиду взаимодействия, проводимости рёбер определяются текущим образом из соотношений:  $f_d(v_i, v_j) = a_{ij}(r_j + q_j)$ ,  $f_c(v_j, v_i) = b_{ij}(r_j + q_j)$ , где  $0 \leq a_{ij} \leq 1$ ,  $0 \leq b_{ij} \leq 1$  — постоянные взаимодействия, определяемые по формулам  $a_{ij} = (a_i a_j)^{0.5}$ ,  $b_{ij} = (b_i b_j)^{0.5}$ , где, в свою очередь,  $0 \leq a_i \leq 1$ ,  $0 \leq b_i \leq 1$  — постоянные мотиваций взаимодействия в каждой вершине  $v_i$  ( $i, j = 0, 1, 2$ ). Для выявления автономных качеств системы все  $a_i, b_i$  и  $0 \leq \lambda_i \leq 1$  считаются неизменными в конкретном ПИФ. Кроме того, ограничимся случаями, когда все  $a_i$  одинаковы и равны  $a$ , также все  $b_i = b$ .

Итак, фиксируя пять значений  $\lambda_0, \lambda_1, \lambda_2, a, b$ , прослеживаем, какой при них складывается режим ПИФ триады. Затем меняем независимо эти значения и отслеживаем, как это влияет на режим ПИФ. Выберем минимум здоровых вариаций. Так, величина  $\lambda_0 + \lambda_1 + \lambda_2$  характеризует инерционность, устойчивость к внешним влияниям, поэтому во всех вариантах она должна быть близка к 1. Далее, спутник менее массивен, чем основное тело, значит, должно быть  $\lambda_0 \leq \lambda_1$ . Наконец, значения  $a$  и  $b$  задают скорость перераспределения ресурса, так что характер режима определяется, в основном, отношением  $a/b$ , при этом факту сплочённости (вязкости) системы соответствует условие  $a < b$ .

Ввиду сказанного, у отправного варианта актуальны значения:  $\lambda_0 = 0,001$ ,  $\lambda_1 = 0,9$ ,  $\lambda_2 = 0$ ,  $a = 0,1$ ,  $b = 0,9$ . Соответствующая программа выдаёт графики изменения величин актива (ряд 1) и пассива (ряд 2) в тройке вершин  $v_0, v_1, v_2$ , считая сверху (рис. 1). Их дополняют графики типов движения в каждой вершине (рис. 1), где в порядке сверху вниз и от ряда 1 к ряду 2 представлены: равномерное прямолинейное движение и вакуумная ёмкость, механическое и вращательное (поворот) движения, импульс напряжения и тепловое движение, стабильное вращательное (вокруг оси) движение и теплоёмкость [5].

Далее естественно взять случай с «тормозной» средой, где  $\lambda_2 > 0$ . На рис. 2 приведён вариант с заменой  $\lambda_2 = 0$  на  $\lambda_2 = 0,05$ . Чтобы проследить влияние массивности спутника, проводится наращивание



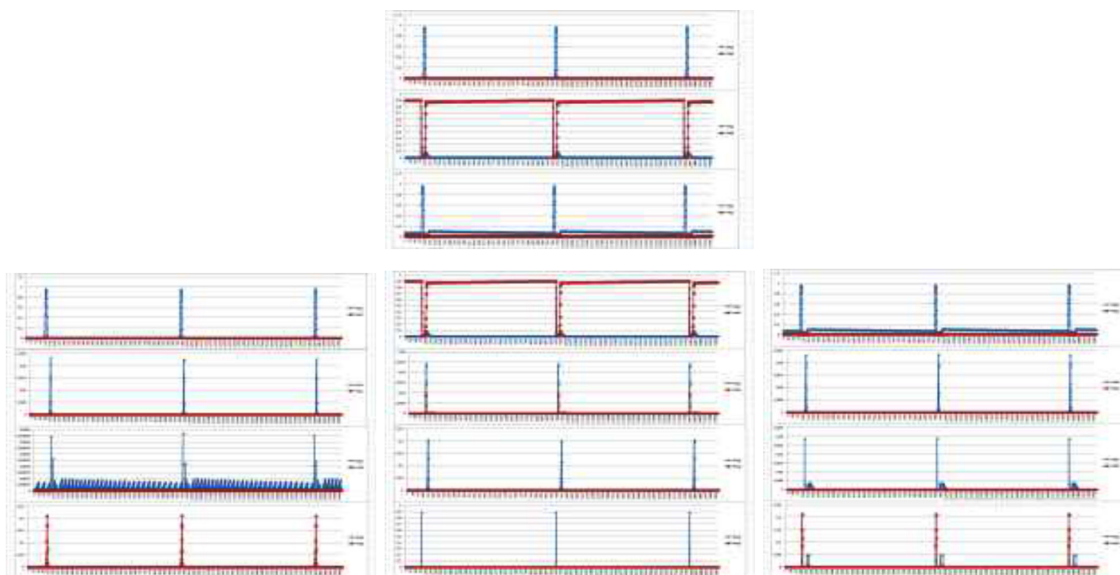


Рис. 2. Преобладание электромагнетизма

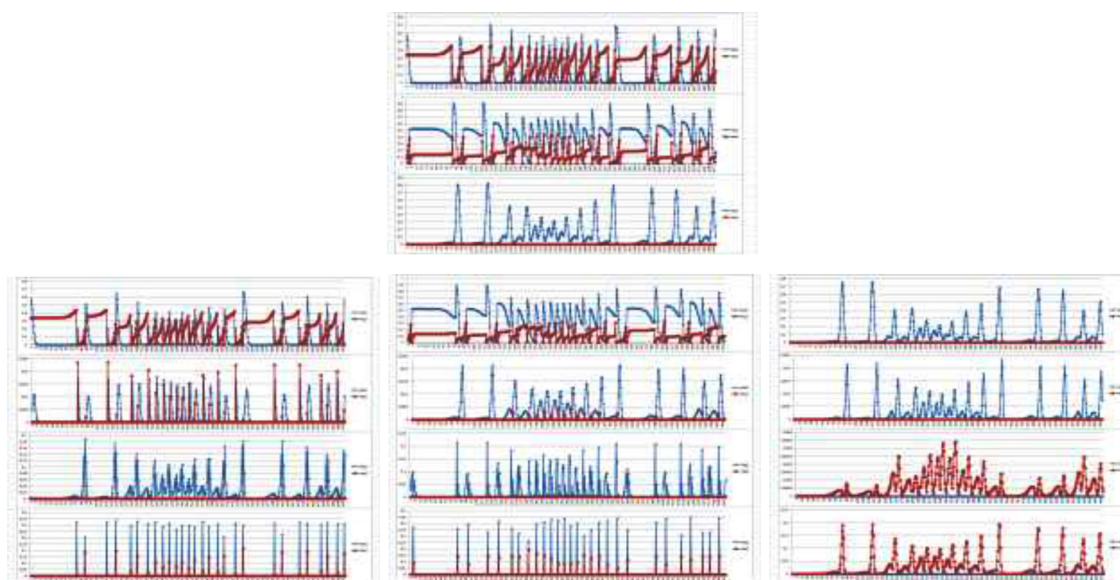


Рис. 3. Гравитация у сопоставимых тел

отношения  $\lambda_0/\lambda_1$  с условием  $\lambda_0 + \lambda_1 = 0,9$ . На рис. 3 приведён вариант с  $\lambda_0 = \lambda_1 = 0,45$ . Для прослеживания перемен при постепенном изменении параметров на рис. 4 приведена серия режимов переходного характера уже без графиков движений. Наконец, на рис. 5 представлены варианты, когда влияние среды сопоставимо с влиянием основного тела, а также случай слабой сплочённости, где  $a > b$ .

**Анализ моделей.** Как видно из рис. 1, ПИФ, в главном, периодичен. При этом у основного тела копится пассив до величины  $\lambda_1$ , достижение которой даёт трансформацию пассива в актив, и основное тело надолго сосредоточивает в своём активе почти весь ресурс системы. Скоро этот актив почти весь быстро передаётся среде, породив в ней «вспышку», а среда почти весь актив быстро передаёт спутнику, представив его как «вспышку». Далее спутник быстро передаёт актив основному телу, пополняя у него пассив до объёма  $\lambda_1$  и актив до объёма около  $1 - \lambda_1$ , и ситуация повторяется. Характерной особенностью такого режима ПИФ является посменная, почти полное перетекание ресурса по циклу

$v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow v_0$ . Такие режимы относятся к классу режимов гравитации [8–12].

Обратимся к аддитивным составляющим ПИФ в ранге типов движения. Так, всюду основная часть актива в ПИФ выражает равномерное прямолинейное движение (ряд 1 в первой полосе), т.е. как бы свободное плавание элементов в системе, а основная часть пассива — вакуумную ёмкость (ряд 2 в этой полосе). Механическое движение (ряд 1 во второй полосе) в каждой вершине отражает скорость утекания актива из неё и предстаёт пиками одинаковой небольшой высоты. На фоне этого почти незаметно вращательное движение (ряд 2 во второй полосе), на деле его нет у среды (где  $\lambda_2 = 0$ ), а у спутника оно проявляется почти бесперебойно малыми импульсами — факт плавного движения спутника вокруг основного тела. Импульсы напряжения (ряд 1 в третьей полосе) как натуральные дифференциалы, естественно, малы, выписывая узоры с частыми скачками у спутника и повторяя со сдвигом показания механического движения у основного тела. Тепловое движение



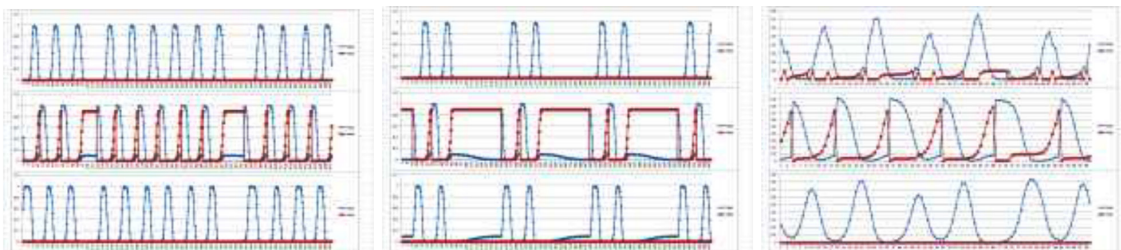


Рис. 4. Хаос периодов (слева), эффект молнии (в центре) и гравитация при относительно массивном спутнике (справа)

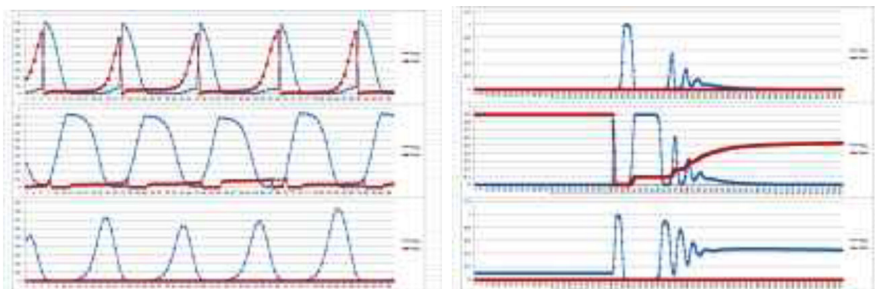


Рис. 5. Гравитация при властном спутнике (слева) и электромагнитные волны

(ряд 2 в третьей полосе) проявляется в такт с механическим, его больше у среды и спутника и почти нет у основного тела. Перегрева и сгорания нет, но это не исключено у спутника при  $a > 0,5$ , когда показатели тепла могут долго преобладать над теплоёмкостью (ряд 2 в четвёртой полосе). Стабильного вращательного движения (ряд 1 в этой полосе) нет у среды, редким и слабым оно имеется у спутника, продиктованное его движением вокруг основного тела, а у основного тела очень резкий импульс к вращению его вокруг оси приходится на момент трансформации в нём пассива в актив.

Таковы особенности проявления процесса гравитационного взаимодействия между телом и его спутником. Теперь обратимся к анализу вариаций.

Постепенный рост значения  $\lambda_2 > 0$  при «тормозной» среде уже при  $\lambda_2 > 0,0005$  даёт хаос продолжительностей «периодов» и амплитуд активов. На рис. 4 слева приведён случай с  $\lambda_2 = 0,0015$ . А при  $\lambda_2 \geq 0,0044$  режим ПИФ вновь становится периодическим, но из класса режимов электромагнетизма [8–12], где основное тело играет роль положительного полюса электрического диполя, спутник — отрицательного, а среда — магнитной среды как сверхпроводника электрического взаимодействия.

На рис. 2 приведён случай с  $\lambda_2 = 0,05$ . Здесь на протяжении почти всего периода основное тело копит пассив до величины  $\lambda_1$ , по достижению которой происходит трансформация пассива в актив. Но теперь этот актив сразу быстро передаётся среде в почти полном объёме, породив в ней «вспышку», за которой скоро следует «вспышка» и у спутника. От спутника актив быстро передаётся основному телу, но это не позволяет телу пополнить свой пассив до объёма  $\lambda_1$ , так как часть ресурса уходит в актив тела, а часть задерживается в пассиве среды, не достигая величины  $\lambda_2$ .

Когда поступающий из основного тела актив приведёт в среде к накоплению пассива до объёма  $\lambda_2$ , свершится в ней трансформация пассива в актив, и появившийся актив начнёт постепенно добираться, в главном, до пассива основного

тела, пока не сбудется величина  $\lambda_1$ . Так ситуация и повторяется в цикле.

В картине движений вновь основная часть показаний актива в ПИФ приходится на равномерное прямолинейное движение, а пассива — на вакуумную ёмкость. Механическое движение снова даёт пики одинаковой небольшой высоты, к которым у основного тела добавляется небольшая «горка». На фоне этого ярко проявляется пик вращательного движения у основного тела как показатель движения тела через среду. Основные пики импульсов напряжения приходятся на момент трансформации пассива в актив у основного тела, к ним у спутника и среды добавляются серии малых пиков, приходящиеся на их собственные трансформации пассива в актив. Тепловое движение заметно проявляется лишь у спутника, но оснований для перегрева и сгорания вновь нет: и  $a < 0,5$ , и теплоёмкость достаточно велика. Стабильное вращательное движение проявляется редко, как и акты трансформации пассива в актив, наиболее слабы они у спутника, наиболее сильны — у основного тела, а у среды, по сути, на смену тепловому движению пришло стабильное вращательное как аналог магнитных спинов в мире элементарных частиц.

Дальнейшее увеличение  $\lambda_2$  будет вести к увеличению периода у режима электромагнетизма. Но при  $\lambda_2 \geq 0,15 > 1 - \lambda_1$  система терпит коллапс, уходит в вакуум, весь её ресурс сосредоточивается в пассиве. Возможно, это факт выпадения спутника на основное тело из-за серьёзного торможения средой.

Несмотря на электромагнетизм, феномен молнии в представленном варианте отсутствует. Для такого феномена важно, чтобы среда была слабо «тормозной», сочетая электромагнетизм и гравитацию. Можно высказать гипотезу, что молния, по сути, представляет факт схлопывания электромагнетизма в гравитацию. Это даёт приведённый на рис. 4 в центре случай с  $\lambda_2 = 0,003$ .

Далее, с наращиванием массивности спутника, т.е. ростом  $\lambda_0$ , график у спутника будет иметь всё

более серьёзные локальные сдвиги из-за накопленных и трансформаций уже приличных объёмов пассива, картина ПИФ будет отличаться от базового режима (рис. 1) всё большим хаосом амплитуд и уменьшением их средних величин. Но феномен посменного перетекания ресурса, в главном, сохраняется, так что по-прежнему уместно говорить о режиме гравитации. На рис. 4 справа приведён случай, когда  $\lambda_0 = 0,1$  (и  $\lambda_1 = 0,8$ ). В какой-то мере этот феномен сохраняется вплоть до случая  $\lambda_0 = \lambda_1 = 0,45$  (рис. 3). При этом, несмотря на величие спутника, у основного тела сохраняется явное превосходство в показателях актива, а также спутник больше подвержен вращательному движению и меньше — механическому.

Формально допустимо  $\lambda_0 > 0,45 > \lambda_1$ . Но уже при  $\lambda_0 \geq 0,5$  (и  $\lambda_2 = 0$ ) систему ждёт выход на режим поглощения, где на месте среды вообще не остаётся ресурса, на месте спутника — только пассив, как у чёрной дыры, поглотившей среду, а доля актива сохраняется лишь у основного тела.

Однако  $\lambda_0 \geq 0,5$  не даст поглощения при  $\lambda_2 > 0$ , и на рис. 5 слева приведён случай, где  $\lambda_0 = 0,8$ ,  $\lambda_1 = 0,1$  и  $\lambda_2 = 0,001$ . И ведь картина при этом во многом повторяет вариант с гравитацией, приведённый на рис. 4 справа, как если бы и впрямь в нём поменять ролями спутник и основное тело. И этот факт показывает, что феномен взаимодействия тел обретает симметрию лишь в присутствии «тормозной» среды. Однако проследивается и элемент асимметрии — более медленная в обоих случаях разгрузка «горок» актива у основного тела по сравнению со спутником. Впрочем, с учётом теоремы 25 из [5], факт медленной разгрузки «горки» актива в самом деле ассоциирует с большей массой при этой «горке».

Нельзя, конечно, оставлять без внимания случаи не просто «тормозной», но весьма массивной, плотной среды, когда значение  $\lambda_2$  велико. Здесь надо будет полагать  $\lambda_0 = 0,001$  и  $\lambda_1 + \lambda_2 = 0,9$ . И это повторит уже изложенные режимы гравитации с учётом циклической перестановки вершин триады: спутник вместо среды, основное тело вместо спутника и среда вместо основного тела.

Если обратиться к переменам отношения  $a/b$ , то его уменьшение склоняет ПИФ к режиму гравитации, а увеличение — к электромагнетизму, причём скоро к режиму электромагнитных волн. На рис. 5 справа приведён случай, когда  $\lambda_0 = 0,001$ ,  $\lambda_1 = 0,9$ ,  $\lambda_2 = 0$ , но  $a = 0,9$ ,  $b = 0,1$ . Здесь имеем диполь с переменным электромагнитным полем и почти гарантированно сгорание спутника.

**Основные выводы.** Проведённые исследования приводят к серии выводов.

Во-первых, величина уровня трансформации пассива в актив, несмотря на её власть менять характер режима ПИФ, не всегда находится в прямой корреляции с проявлением массы. Это надёжно реализуется лишь в условиях среды-зомби, которая, однако, несмотря на отсутствие с её стороны помех на передачу ресурса, вовсе не идентична традиционной среде в ранге вакуума. Возможно, такая среда и есть поле тяготения. А в условиях даже слегка «загрязнённой» среды, в принципе, любое из двух тел может оказаться спутником другого, если, конечно, нет заранее установленного взаимодействия с другими телами.

Во-вторых, при отсутствии большого контраста между мотивациями пассива и актива режим электромагнетизма сбывается при наличии особо власт-

ного элемента в системе, в противном случае верх берёт режим гравитации.

В-третьих, режим гравитации надёжнее и устойчивее проявляется при росте сплочённости (вязкости) системы, когда пассив мотивирован больше чем актив. И, наоборот, ослабление сплочённости системы, когда актив мотивирован больше чем пассив, способствует проявлению режима электромагнетизма с преобладанием электромагнитных волн. Существование монополий электрических зарядов продиктовано не практикой, а традицией считать  $b = 0$ .

В-четвёртых, риски разогрева и сгорания у спутника вероятны при мотивации актива  $a > 0,5$ . Плотная среда способствует перегреву основного тела, но не спутника. В свою очередь, достаточно серьёзно «загрязнённая» среда обеспечивает торможение и выпадение спутника на основное тело.

В-пятых, выявленные особенности относятся к взаимодействию не только небесных тел. Они справедливы в любых масштабах — от мегадо микромира.

**Закключение.** Фундаментальный потенциал, заложенный в задаче взаимодействия двух тел, становится возможным раскрывать уже с простейших вариантов её системного анализа на базе ДИС-технологии.

Уже первые апробации свидетельствуют о том, что среда, в которой взаимодействуют тела, вовсе не является вакуумом в традиционном понимании. По крайней мере, эта среда не пуста на информационном уровне, выражая, возможно, сущность поля тяготения. А феномен массы обретает смысл лишь в процессе взаимодействия тел, и до вступления во взаимодействие в условиях «замусоренной» среды невозможно предсказать, как два тела поделят между собой роли основного тела и спутника.

Феномены гравитации и электромагнетизма не зависят от масштабов системы двух взаимодействующих тел. Решающую роль в выходе на электромагнетизм играет эффект централизации власти в системе. Этому выходу способствует также рост подвижности субстрата в системе, ослабление её сплочённости, выражающейся в отношении показателя мотивации пассива к показателю мотивации актива:  $b/a$ .

Перегрев и сгорание тел стимулируются не «загрязнением» среды, а высокой мотивацией актива:  $a > 0,5$ .

Многообразие результатов и особенностей у задачи взаимодействия двух тел может существенно расширяться при вариантах, когда показатели мотиваций окажутся свои у каждого элемента системы. В частности, здесь одно из тел может быть, например, твёрдым, а другое жидким. Но об этом речь пойдет уже в следующих работах.

#### Библиографический список

1. Сизиков, В. П. Конфигурация приоткрывает завесы в физике [Текст] / В. П. Сизиков // Омский научный вестник. — 2003. — № 4 (25). — С. 74–78.
2. Сизиков, В. П. Рациональный инструмент отражения принципа причинности [Текст] / В. П. Сизиков // Омский научный вестник. — 2005. — № 4 (33). — С. 92–96.
3. Агафонов, А. Л. Язык имитационного моделирования на базе ТДИС в обеспечении качества автоматизации [Текст] / А. Л. Агафонов, В. И. Разумов, В. П. Сизиков // Омский научный вестник. Сер. Приборы, машины и технологии. — 2009. — № 3 (83). — С. 241–245.

4. Сизиков, В. П. К имитационному моделированию на базе ДИС-технологии [Текст] / В. П. Сизиков // Омский научный вестник. Сер. Приборы, машины и технологии. — 2010. — № 1 (87). — С. 186–190.

5. Основы моделирования процессов с языком ДИС-технологии [Текст] : отчёт по НИОКР / В. П. Сизиков, Т. Ю. Круковская, В. Ф. Кузнецов // Математическое моделирование процессов с выходом на имитацию. — Омск : ОмГУПС, 2014. — 121 с. — № 01201368699.

6. Сизиков, В. П. Где, когда и какая возможна динамика систем? [Текст] / В. П. Сизиков // Динамика систем, механизмов и машин. — 2014. — № 3. — С. 231–234.

7. Разумов, В. И. Основы теории динамических информационных систем [Текст] / В. И. Разумов, В. П. Сизиков. — Омск : ОмГУ, 2005. — 212 с.

8. Сизиков, В. П. Модель и анализ взаимодействия протона и электрона на базе ДИС-технологии [Электронный ресурс] / В. П. Сизиков // Теоретические и прикладные аспекты современной науки : сб. науч. тр. по материалам VI Междунар. науч.-практ. конф. В 6 ч. Ч. 3. — Белгород : ИП Петрова М. Г., 2015. — Режим доступа : <http://www.issledo.ru> (дата обращения: 11.03.2015).

9. Сизиков, В. П. Турбулентность: системный статус и управление [Электронный ресурс] / В. П. Сизиков // Теоретические и прикладные аспекты современной науки : сб. науч. тр. по материалам IX Междунар. науч.-практ. конф. В 6 ч. Ч. 2. — Белгород : ИП Петрова М. Г., 2015. — Режим доступа : <http://www.issledo.ru> (дата обращения: 10.05.2015).

10. Сизиков, В. П. Трещины и дислокации: системный статус и управление [Электронный ресурс] / В. П. Сизиков // Современные тенденции развития науки и технологий : сб. науч. тр. по материалам II Междунар. науч.-практ. конф. В 7 ч. Ч. 2. — Белгород : ИП Ткачёва Е. П., 2015. — Режим доступа : <http://www.issledo.ru> (дата обращения: 15.06.2015).

11. Сизиков, В. П. Системный анализ совершенства мышления [Электронный ресурс] / В. П. Сизиков, Т. Ю. Круковская // Современные тенденции развития науки и технологий : сб. науч. тр. по материалам III Междунар. науч.-практ. конф. В 6 ч. Ч. 5. — Белгород : ИП Ткачёва Е. П., 2015. — Режим доступа : <http://www.issledo.ru> (дата обращения: 02.07.2015).

12. Сизиков, В. П. Системный анализ модернизаций в обществе [Электронный ресурс] / В. П. Сизиков, В. И. Разумов // Современные тенденции развития науки и технологий : сб. науч. тр. по материалам IV Междунар. науч.-практ. конф. В 6 ч. Ч. 3. — Белгород : ИП Ткачёва Е. П., 2015. — Режим доступа : <http://www.issledo.ru> (дата обращения: 12.08.2015).

**СИЗИКОВ Виктор Петрович**, кандидат технических наук, доцент (Россия), доцент кафедры высшей математики.

Адрес для переписки: [v\\_p\\_sizikov@mail.ru](mailto:v_p_sizikov@mail.ru)

Статья поступила в редакцию 15.09.2015 г.

© В. П. Сизиков

УДК 004:[004.02+004.588]+377

**А. Е. УЛЬТАН**

Финансовый университет  
при Правительстве РФ, Омский филиал

## ПРОЕКТИРОВАНИЕ ПРОЦЕДУР, УДОВЛЕТВОРЯЮЩИХ U-ТРЕБОВАНИЯМ, ДЛЯ АЛГОРИТМОВ, ОТДЕЛИМЫХ ОТ ЗНАНИЙ

Статья посвящена проектированию интеллектуальных компьютерных компонентов тройного назначения. Она содержит описание архитектуры процедур, отделимых от знаний.

**Ключевые слова:** компонент, образование, Интернет.

Что такое интеллектуальные компьютерные компоненты тройного назначения, удовлетворяющие U-требованиям. На наш взгляд [1–4], необходимо создавать, способные взаимодействовать между собой интеллектуальные компьютерные компоненты тройного назначения (название наше), выводящие в Интернет различные предметные области с задачами. Внешне, для удобства пользователей, они все, на наш взгляд, должны выглядеть единообразно, т. е. удовлетворять следующим требованиям, которые мы назвали U-требования:

а) они должны содержать как минимум UML-учебник (наше понятие), приспособленный для единообразного компьютерного системного изложения материала;

б) они должны уметь самостоятельно решать предложенные задачи и на основе этого предостав-

лять и объяснять решение, следить за решением человека и локализовать ошибки, подсказывать ход решения намеками или сделав один шаг решения. Допущенные ошибки должны фиксироваться для генерации в будущем задач с подобными вызвавшими затруднения ситуациями.

Умение самостоятельно решать задачи должно поддерживаться самообучаемостью компонентов, заключающейся в том, что задачи, которые компонент не может решить, передаются человеку, чтобы он показал ход решения, а компонент выделил в нем и показал человеку те переходы, которые он делать не умеет. Если человек после этого перепрограммирует компонент, то тот сможет в дальнейшем справляться с подобными моментами.

Эффективности обучения должна способствовать возможность компонентов обращаться друг

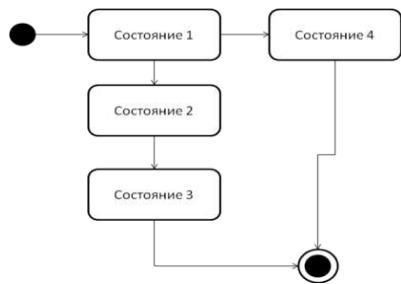


Рис. 1. Пример диаграммы состояний

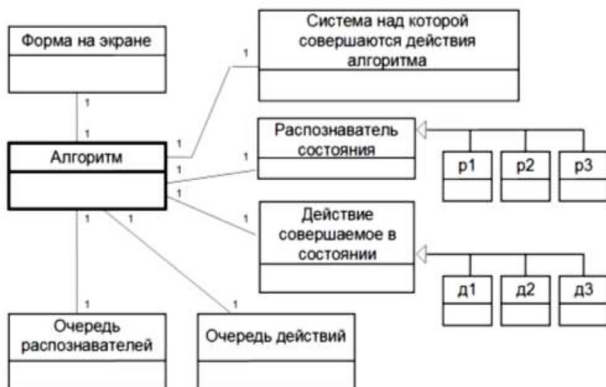


Рис. 2. Диаграмма классов для алгоритма отделимого от знаний

к другу через Интернет для выполнения каждым из них той части решения, на которой он специализируется. При этом взаимодействие компонентов должно быть таким, чтобы все педагогические функции сопровождали и этапы решения подзадач. Тогда во время решения, например, задачи по физике будут стандартно обработаны также ошибки по алгебре, арифметике и т. д.;

в) умение самостоятельно решать задачи, делать следствия и доказательства должно предоставляться не только учащимся и учителям, но также программистам, а в перспективе и ученым. Именно поэтому мы и назвали такие компоненты компонентами тройного назначения.

**Что такое алгоритм, отделимый от знаний.** Примером алгоритма, отделимого от знаний, является алгоритм решения квадратного уравнения. Многие знают, какие действия и в какой взаимосвязи необходимо осуществлять на каждом шаге, но не многие смогут объяснить, доказать, почему это так, потому что в этом случае объяснения, доказательства каждого действия в отдельности объединяют в объяснение, доказательство всего алгоритма. Поэтому тут можно совершать отдельно описание совокупности необходимых действий и описание доказательства необходимости и достаточности этих действий. Употреблять эти описания можно отдельно. Именно в этом смысле алгоритм является отделимым от знаний.

**Проектирование процедур, удовлетворяющих У-требованиям для алгоритмов, отделимых от знаний.** Допустим, что условие задачи описывает некоторую исходную, стартовую систему (например, задачу линейного программирования), которую некоторый алгоритм, отделимый от знаний, переводит в некоторую итоговую, финишную систему (ответ). В этом случае мы заранее знаем все действия ал-

горитма и поэтому можем описать его с помощью диаграммы состояний из языка UML (Universal modeling language), изображенной на рис. 1.

Каждое состояние характеризуется некоторой подцелью (она дает имя состоянию), действиями, которые необходимо выполнить над системой для достижения подцели, и стрелками, указывающими, в какое новое состояние необходимо перейти. Если после выполнения всех действий система удовлетворяет условию, расположенному рядом со стрелкой, или в любом случае, если условие возле стрелки отсутствует. Возможен переход из состояния в самое себя. Каждое состояние может содержать подсостояния. Тогда среди них должно быть начальное подсостояние, в которое мы попадем, войдя в состояние и конечное подсостояние. Приход в конечное подсостояние означает, что все действия его надсостояния выполнены и необходимо перейти из надсостояния в другое состояние согласно выходящим из надсостояния стрелкам. Однако до конечного подсостояния мы можем не дойти, т. к. допускается переход из внутреннего подсостояния за пределы надсостояния в другое состояние или его подсостояние.

Так как состояния могут быть не только связаны стрелками-переходами, но и могут образовывать древовидную иерархию за счет связей надсостояние-подсостояние (будем считать, что самое верхнее состояние-вершина дерева имеет имя алгоритма, которому соответствует диаграмма состояний), то каждое состояние наряду с просто именем будет обладать и составным именем, содержащим перечисление через точку имен состояний-вершин дерева, по которым необходимо пройти из самого верхнего состояния в интересующее нас состояние.

Каждому состоянию мы поставим в соответствие несколько классов: класс-распознаватель следующего состояния с процедурой «Выполнить» и по одному классу-действию с процедурой «Выполнить» для каждого действия состояния.

Тогда наш алгоритм можно единообразно описать с помощью укрупненной диаграммы классов, изображенной на рис. 2.

Из диаграммы видно, что алгоритм ассоциируется с исходной системой, над которой будут совершаться действия, с действиями, которые в соответствующих состояниях можно произвести над системой и с распознавателями будущих состояний.

Класс «Алгоритм»-управляющий. Его конструктор (процедура, запускаемая, когда создается объект класса «Алгоритм») делает следующее:

- создает исходную систему, которую будет преобразовывать алгоритм (например, пустую симплекс-таблицу, если речь идет об алгоритме решения задачи линейного программирования);

- создает форму, изображающую меню и элементы системы для общения с пользователем (задания условия задачи и ее решения);

- создает объект «Состояние ученика». Первоначально устанавливается состояние «Задание условия задачи». Но после нажатия пользователем на кнопку «Самостоятельное решение» объект перейдет в состояние «Самостоятельное решение». После нажатия пользователем на кнопку «Автоматическое решение» перейдет в состояние «Автоматическое решение». После нажатия на кнопку «Сделай шаг» перейдет в состояние «Сделай шаг». После нажатия на кнопку «Намекни» перейдет в состояние «Намекни». После выполнения намека или действий одного шага объект «Состояние



ученика» самостоятельно возвращается в состояние «Самостоятельное решение», а из состояния «Автоматическое решение» не возвращается, и решение производится до конца. После окончания решения задачи доступными остаются состояния «Задание условия задачи», возможно «Анализ решения» и «Завершение работы»;

— создает стек «Очередь распознавателей» и стек «Очередь действий».

В вершине стека «Очередь распознавателей» всегда будет располагаться распознаватель будущих состояний того текущего состояния, в котором мы находимся в данный момент.

В стек «Очередь действий» всегда помещается цепочка действий, соответствующая текущему на данный момент состоянию (т. е. состоянию, распознаватель которого находится в данный момент в вершине стека «Очередь распознавателей»). В самом вершине стека помещается первое действие, ниже — следующее, и т. д.

Если же в текущем состоянии ничего не требуется делать, то в стек «Очередь действий» ничего не помещается, и он будет пуст. Помещать свою цепочку действий в стек «Очередь действий» будет текущий распознаватель после своего появления в вершине стека «Очередь распознавателей».

Первоначально в стек «Очередь распознавателей» управляющий объект «Алгоритм» — помещает распознаватель начального состояния, а в стек «Очередь действий» ничего не помещает (в начальных состояниях делать ничего не надо. Из них необходимо лишь идти дальше согласно стрелкам). Затем управляющий объект «Алгоритм» начинает циклически вызывать процедуру «Выполнить» текущего распознавателя до тех пор, пока стек «Очередь распознавателей» не окажется пустым.

Процедура «Выполнить» любого распознавателя работает так: сначала она начинает циклически вызывать процедуру «Выполнить» действия находящегося в вершине стека «Очередь действий» и удалять это действие после выполнения. Делается это, пока стек «Очередь действий» не опустеет. Получаем объект «Система, над которой совершаются действия алгоритма» с элементами, над которыми выполнены все действия, которые необходимо сделать в текущем состоянии. При этом каждое действие после преобразования некоторого элемента оставляет в нем запись о том, что было сделано.

С другой стороны, к каждому элементу подключается процедура-обработчик события Click. Эта процедура заработает тогда, когда ученик щелкнет мышкой по элементу системы, к которому она подключена. Запускаемая таким образом процедура выдаст хранящееся в элементе системы сообщение о том, как он был получен.

Затем, если пользователь находится в состоянии «Автоматическое решение», осуществляется переход в новое состояние, описанное ниже. При этом, если переход помеченный, то пользователю выдается как промежуточный результат объект «Система, над которой совершаются действия алгоритма», если нет, то это не делается.

Если же пользователь находится в состоянии «Самостоятельное решение», то ему предоставляется возможность преобразовывать элементы системы самостоятельно. Каждый раз полученное пользователем сравнивается с правильно преобразованным элементом и в случае ошибки выдается сообщение об этом.

При этом, если пользователь перейдет в состояние «Намекни», то ему будет указано, в каком состоянии он находится, и выдан намек на возможность преобразования тех элементов, которые пользователем еще не преобразованы.

Если пользователь перейдет в состояние «Сделай шаг», то ему будет указано, в каком состоянии он находится, и будет показано преобразование одного из тех элементов, которые пользователем еще не преобразованы.

Если же пользователь перейдет в состояние «Автоматическое решение», то решение будет продолжено автоматически до конца.

Когда пользователь, находясь в состоянии «Самостоятельное решение», правильно преобразует все нужные элементы, осуществляется переход в новое состояние. Если переход помеченный, то пользователю выдается как промежуточный результат объект «Система, над которой совершаются действия алгоритма», если нет, то это не делается.

Переход в новое состояние осуществляется так:

— если мы переходим в некоторое **начальное подсостояние** некоторого надсостояния, то:

— вместо распознавателя текущего состояния в стеке «Очередь распознавателей» добавляется сначала распознаватель надсостояния, а затем распознаватель начального подсостояния. В стек «Очередь действий» ничего не добавляется;

— если мы переходим в некоторое **состояние или подсостояние**, то:

— если мы переходим в **некоторое состояние или подсостояние не выходя из надсостояния**, то вместо распознавателя текущего состояния в вершину стека «Очередь распознавателей» добавляется распознаватель нового состояния. В стек «Очередь действий» сверху добавляются действия нового состояния, если такие имеются.

Если же мы переходим в некоторое состояние или подсостояние, выходя из надсостояния, то стек «Очередь распознавателей» сначала очищаем, а затем заполняем распознавателями состояний, имена которых присутствуют в полном имени состояния, в которое мы перешли. Состояние, в которое мы перешли, получится в вершине стека «Очередь распознавателей». Затем стек «Очередь действий» заполняется действиями нового текущего состояния или остается пустым, если в этом состоянии делать ничего не надо;

— если мы переходим в **конечное состояние или подсостояние**, то:

— убираем распознаватель текущего состояния из стека «Очередь распознавателей». Стек «Очередь действий» заполняется действиями нового состояния оказавшегося в вершине стека «Очередь распознавателей» или остается пустым, если в этом состоянии делать ничего не надо.

Ясно, что в данной работе описана лишь общая схема проектирования процедур, отделимых от знаний.

В конкретных ситуациях необходимо вносить дополнения и изменения, соответствующие ситуации.

При этом эти изменения и дополнения могут оказаться очень интересными, но об этом — в других работах.

#### Библиографический список

1. Ультан, А. Е. Разработка обучающей программы «Решение текстовых задач по алгебре» / А. Е. Ультан, В. А. Зakan-

дырин // Омский научный вестник. Сер. Приборы, машины и технологии. — 2009. — № 6 (82). — С. 200–201.

2. Ультан, А. Е. Разработка архитектуры комплекса обучающих программ / А. Е. Ультан, Е. С. Петров // Омский научный вестник. Сер. Приборы, машины и технологии. — 2010. — № 6 (92). — С. 190–193.

3. Ультан, А. Е. Разработка обучающей информационной системы «Алгебра» / А. Е. Ультан, Д. А. Кравцов // Омский научный вестник. Сер. Приборы, машины и технологии. — 2011. — № 3 (98). — С. 169–173.

4. Ультан, А. Е. Что такое «Интеллектуальные компьютерные компоненты тройного назначения» и основные факторы,

влияющие на их устройство / А. Е. Ультан // Омский научный вестник. Сер. Приборы, машины и технологии. — 2015. — № 3 (143). — С. 287–289.

**УЛЬТАН Александр Ефимович**, кандидат технических наук, доцент кафедры «Информатика и информационные технологии».

Адрес для переписки: ultan\_ae@mail.ru

Статья поступила в редакцию 25.05.2015 г.

© А. Е. Ультан

УДК 004:[004.02+004.588]+377

**А. Е. УЛЬТАН**

**Финансовый университет  
при Правительстве РФ, Омский филиал**

## **ПРОЕКТИРОВАНИЕ ПРОЦЕДУР, УДОВЛЕТВОРЯЮЩИХ U-ТРЕБОВАНИЯМ, ДЛЯ АЛГОРИТМОВ, НЕ ОТДЕЛИМЫХ ОТ ЗНАНИЙ**

**Статья посвящена проектированию интеллектуальных компьютерных компонентов тройного назначения. Она содержит описание архитектуры процедур, не отделимых от знаний.**

**Ключевые слова:** компонент, образование, Интернет.

Что такое интеллектуальные компьютерные компоненты тройного назначения, удовлетворяющие U-требованиям? Согласно концепции, изложенной в [1–4], необходимо создавать способные взаимодействовать между собой интеллектуальные компьютерные компоненты тройного назначения (название наше), выводящие в Интернет различные предметные области с задачами. Внешне, для удобства пользователей, они все, на наш взгляд, должны выглядеть единообразно, т. е. удовлетворять следующим требованиям, которые мы назвали U-требования:

а) они должны содержать как минимум UML-учебник (наше понятие), приспособленный для единообразного компьютерного системного изложения материала;

б) они должны уметь самостоятельно решать предложенные задачи и на основе этого предоставлять и объяснять решение, следить за решением человека и локализовать ошибки, подсказывать ход решения намеками или сделав один шаг решения. Сделанные ошибки должны фиксироваться для генерации в будущем задач с подобными вызвавшими затруднения ситуациями.

Умение самостоятельно решать задачи должно поддерживаться самообучаемостью компонентов, заключающейся в том, что задачи, которые компонент не может решить, передаются человеку, чтобы он показал ход решения, а компонент выделил в нем и показал человеку те переходы, которые он делать не умеет. Если человек после этого перепрограммирует компонент, то тот сможет в дальнейшем справиться с подобными моментами.

Эффективности обучения должна способствовать возможности компонентов обращаться друг к другу через Интернет для выполнения каждым из них той части решения, на которой он специализируется. При этом взаимодействие компонентов должно быть таким, чтобы все педагогические функции сопровождали и этапы решения подзадач. Тогда во время решения, например, задачи по физике будут стандартно обработаны также ошибки по алгебре, арифметике и т. д.;

в) умение самостоятельно решать задачи, делать следствия и доказательства должно предоставляться не только учащимся и учителям, но также программистам, а в перспективе и ученым. Именно поэтому мы и назвали такие компоненты компонентами тройного назначения.

**Что такое алгоритм, не отделимый от знаний.** Когда мы решаем задачу по геометрии, или упрощаем алгебраическое выражение, или решаем уравнение и т. д., то мы заранее известную совокупность шагов, решающую задачу, не знаем. Здесь мы действуем пошагово, но должны с помощью знаний обосновать допустимость каждого шага отдельно (однако эта допустимость еще не означает необходимость). Так мы поступаем либо пока не получим ответ, либо пока не откажемся от попыток решить задачу.

Получив ответ, мы можем отделить все шаги, необходимые для решения, от других допустимых, но не необходимых шагов. Как видим, при использовании алгоритма, не отделимого от знаний, мы на каждом шаге должны использовать знания, позволяющие увидеть и обосновать возможность



Рис. 1. Общий вид диаграммы классов для проектирования процедур, удовлетворяющих U-требованиям, для алгоритмов, не отделимых от знаний

некоторого действия. Именно в этом смысле алгоритм является неотделимым от знаний.

**Проектирование процедур, удовлетворяющих U-требованиям, для алгоритмов не отделимых от знаний.** В этом случае мы не знаем заранее действий алгоритма, нужным образом преобразующего стартовую систему (например, условие задачи) в финишную систему, удовлетворяющую нужным требованиям (например, ответ). Наша задача найти с помощью знаний действия такого алгоритма.

Несмотря на разнообразие подобных ситуаций, всех их объединяет подход, изображенный на рис. 1.

На этом рисунке изображен самый общий вид диаграммы классов для проектирования процедур, удовлетворяющих U-требованиям для алгоритмов, не отделимых от знаний.

Класс «Алгоритм» — управляющий. Его конструктор (процедура, запускаемая, когда создается объект класса «Алгоритм») делает следующее:

- устанавливает объект «Состояние ученика» в состояние «Самостоятельное решение»;
- создает исходную систему, которую будет преобразовывать алгоритм;
- создает форму для общения с пользователем, изображающую элементы системы;
- для фиксирования появившихся возможностей сделать определенные действия создается «Система действий».

Как видно из рис. 1, все действия нужно располагать послыно таким образом, чтобы каждое новое действие содержало ссылки на предшествующие действия и на последующие действия, и при этом чтобы номер слоя последующего действия был всегда больше номеров слоев, содержащих предшественников. Такой послынный процесс продвижения вперед назовем «прямым ходом»;

— создаются источники следствий, некоторые из которых обладают определенными знаниями о системе и способны распознать, что если в системе сложилась определенная ситуация, на которой специализируется этот источник следствий, то с системой можно произвести соответствующее действие.

Другие источники следствий способны обнаружить, что система после некоторого действия пришла в нужное финишное состояние. Однако обнаружение этого факта не означает, что процесс преобразования системы должен быть прекращен.

Напротив, он должен быть продолжен для обнаружения всех возможных способов преобразования исходной системы в финишную систему, обладающую нужными свойствами. Все такие способы преобразования можно найти с помощью «обратного хода». Таким образом, с помощью «обратного хода» можно отделить необходимые действия от достаточных действий, а кроме того, выделить самый простой способ преобразования исходной системы в финишную систему.

Выполнение процедуры зависит от того, в каком состоянии находится объект «Состояние пользователя»:

- каждый раз, когда объект «Состояние ученика» устанавливается в состояние «Самостоятельное решение», пользователю предоставляется возможность самостоятельно делать те или другие действия. Выполненное действие будет оцениваться на предмет возможности в данной ситуации;
- в состоянии «Сделай шаг» процедура выполняет одно из возможных действий и отмечает в карточке ученика, что он не владеет знаниями тех источников следствий, которые приводят к возможным в данной ситуации действиям. Объект «Состояние ученика» возвращается в состояние «Самостоятельное решение»;
- в состоянии «Автоматическое решение» процедура выполняется так же, как и в состоянии «Сделай шаг». Отличие в том, что после выполнения действия возвращения в состояние «Самостоятельное решение» не происходит;
- в состоянии «Намекни» процедура выводит намек на возможные в данной ситуации действия. Объект «Состояние ученика» возвращается в состояние «Самостоятельное решение».

После выполнения всех действий, как отмечалось выше, с помощью «обратного хода» находится и помещается в коллекцию «Ход решения» самый простой способ преобразования исходной системы в финишную систему.

Затем запускается процедура «Выдать решение», которая на основе коллекции «Ход решения» выдает все решение или часть решения, полученную в состоянии «Автоматическое решение».

Процедура «Выдать решение» очень важна. С одной стороны, она должна выдавать решение не очень подробно, чтобы не раздражать хороших учеников. А с другой — работая с выданным решением, ученики должны иметь возможность получить

любые самые подробные объяснения с помощью щелчка мышкой на значке между преобразованными вариантами системы, для вызова окна «Так как», содержащего всю необходимую информацию.

#### Библиографический список

1. Ультан, А. Е. Разработка обучающей программы «Решение текстовых задач по алгебре» / А. Е. Ультан, В. А. Зakanдырин // Омский научный вестник. Сер. Приборы, машины и технологии. — 2009. — № 6 (82). — С. 200–201.
2. Ультан, А. Е. Разработка архитектуры комплекса обучающих программ / А. Е. Ультан, Е. С. Петров // Омский научный вестник. Сер. Приборы, машины и технологии. — 2010. — № 6 (92). — С. 190–193.
3. Ультан, А. Е. Разработка обучающей информационной системы «Алгебра» / А. Е. Ультан, Д. А. Кравцов // Омский

научный вестник. Сер. Приборы, машины и технологии. — 2011. — № 3 (98). — С. 169–173.

4. Ультан, А. Е. Что такое «Интеллектуальные компьютерные компоненты тройного назначения» и основные факторы, влияющие на их устройство / А. Е. Ультан // Омский научный вестник. Сер. Приборы, машины и технологии. — 2015. — № 3 (143). — С. 287–289.

**УЛЬТАН Александр Ефимович**, кандидат технических наук, доцент кафедры «Информатика и информационные технологии».

Адрес для переписки: ultan\_ae@mail.ru

Статья поступила в редакцию 25.05.2015 г.

© А. Е. Ультан

УДК 519.7

**А. Н. ФЛОРЕНСОВ**

Омский государственный  
технический институт

## О МАТЕМАТИЧЕСКОЙ МОДЕЛИ РАЗВИТИЯ РАЗУМА

**На основе анализа феномена феральных людей установлено, что разум, как естественнонаучное понятие, представляет собой не качественную характеристику, а должен описываться количественной величиной и его формирование с систематическим увеличением этой величины происходит исключительно в результате последовательности взаимодействий с другими индивидуальными разумами. Представлена математическая модель формирования и развития разума, основанная на информационных взаимодействиях. Выявлены составляющие аналитической величины приращения величины разума в одиночных взаимодействиях, установлена роль индивидуального и социального фактора в них с получением зависимости, которая может использоваться в образовании для эффективного дальнейшего развития разума.**

**Ключевые слова:** разум, интеллект, величина разума, взаимодействие, информация, математическая модель.

**Введение.** Разум, как объект воспринимаемый, прежде всего, «изнутри», для человека настолько привычен, что рассматривается преимущественно с эмоциональных позиций. При взгляде на него со стороны он является скорее предметом философских дискуссий, чем естественнонаучного исследования. Сложность подобного исследования определяется, в первую очередь, как чрезвычайным многообразием всевозможных проявлений разума, так и почти полным отсутствием инструментов такого исследования.

Широко распространенное убеждение о человеческом разуме, как воспроизводимом в человеческом обществе самопроизвольно, по своим собственным «объективным законам», не более чем иллюзия. Это утверждение легко доказывается феноменом феральных людей, как называют людей, воспитанных только животными. Известных фактов воспитания животными человеческих детей накопилось и изучено многие сотни [1], и все они свидетельствуют, что если в формировании человека систематически не участвовал хотя бы один человек, то из ребенка вырастает организм с жи-

вотными навыками, но совершенно без признаков разума. Отсюда вытекает *необходимость систематического взаимодействия на формируемый человеческий разум других разумных людей*, а для науки — изучения управляющих условий формирования разума. Ближе к этому стоит и общечеловеческая проблема создания условий воспитания разума в реальном процессе человеческого обучения и образования.

**1. Разум и интеллект.** Следует различать понятие человеческого разума, как общее для философии и антропологии, от более узкого и специфического для психологии понятия «интеллект». Под интеллектом, как научным термином, понимают объединение познавательных способностей человека [2]. Разумом, согласно определению, данному еще В. С. Соловьевым [3], называют «существенную для человека, как вида Homo Sapiense, способность *мыслить всеобщее* в отличие от непосредственно данных единичных фактов, какими исключительно занято мышление прочих животных». Разум по существу своему дает то самое существовавшее свойство, которое выделяет род людской



из мира животных. Интеллект же более детально пытаются детализировать некоторые моменты и особенности этого свойства. На основе понятий кибернетики автором данной работы ранее было предложено краткое определение разума как «системы моделирования окружающего мира и адаптации к его воздействиям с помощью этих моделей» [4].

Понятие интеллекта стало широко использоваться с XX века для попыток количественного измерения умственных способностей человека. На этом пути возникло понятие «коэффициент интеллекта», которое дает количественную оценку уровня интеллекта человека, но только относительно уровня интеллекта среднестатистического человека такого же возраста. Эта количественная оценка формируется с помощью специальных тестов. Практически тесты, как они сложились в области психологии, требуют навыков работы с текстами языка, на котором написаны. Необходимым условием их применения являются навыки, полученные в современном школьном образовании, особенно навыки работы с письменными текстами. Они также требуют определенной дисциплины выполнения заданий, поэтому применение чего-то подобного для ребенка малых лет оказывается непригодным.

**2. Разум как количественная характеристика человека.** Поскольку в основе формирования разума лежат взаимодействия между людьми, то именно с опорой на описание и объективное оценивание таких взаимодействий должны строиться методы и формальные средства оценивания и описания процесса формирования разума и его текущих показателей, доступных в той или иной мере для измерения, оценивания, моделирования и вычисления.

Разум не может быть только качественной характеристикой человека, иначе должны быть отчетливо проявляемые события последствий перехода из исходного состояния неразумности человека в состояние его разумности. Вся человеческая практика воспоминаний и восприятия других людей свидетельствует — никакой резкой или сколько-нибудь ощутимой временной границы между отсутствием разума и его наличием наблюдать не удается.

Отсюда, с научной точки зрения, вытекает существование некоторой, пусть в практике и явно не вычисляемой, количественной величины разума. По существу, эта функция служит внешней мерой разума и будет рассматриваться как безразмерная величина. Для формализованных построений обозначим ее функцией  $A(i, t)$ , где параметр  $i$  задает идентификацию конкретного индивидуума, а параметр  $t$  — момент времени текущего значения «величины разума». Для практического и формализованного построения модели с взаимодействием различных разумов целесообразно перейти к общей шкале времени. Параметр  $t$  отсчитывается от общей точки отсчета, в качестве которой можно принять любую из подобных точек отсчета в исторической шкале. Функция  $A(i, t)$  может отличаться от нуля только во время жизни человека, т. е. в интервале от  $t_1$  до  $t_2$ , где  $t_1$  — момент рождения человека, а  $t_2$  — момент его смерти.

**3. Предварительное описание математической модели.** Поскольку текущее значение величины разума есть интегральная функция его взаимодействий с разумами других людей, то функция  $A(i, t)$  определяется как интеграл всевозможных последствий от взаимодействий индивидуума с другими индивидуумами за интервал времени  $[t, t]$ . Все эти

взаимодействия можно считать дискретными, что следует из известной дискретности физиологических нервных взаимодействий, из дискретности передачи по нервным волокнам и реакций синапсов. Поэтому общий интеграл сводится к элементарной сумме конечного ряда последствий отдельных взаимодействий.

Поскольку определяющими для формирования разума являются исключительно человеческие тексты, прежде всего в их звуковой форме, а воспринимать более чем один из одновременно звучащих или просматриваемых текстов оказывается практически невозможным, то можно учитывать только парные взаимодействия индивидуумов.

Отдельный суммируемый для очередного увеличения функции  $A(i, t)$  на интервале  $[t, t + \Delta t]$  элемент приращения  $\Delta A$  индивидуального разума  $i$  может быть описан математически как произведение двух сомножителей  $B(i, t)$  и  $D(i, t)$ . Первый из них является количественной характеристикой интенсивности взаимодействия, а второй выражает потенциальную возможность поступления новых возможностей от другой стороны взаимодействия. Величина  $D(i, t)$  зависит от обеих сторон взаимодействия, фактически это неявная функция от текущего показателя величины разума, как индивидуума  $i$ , так и индивидуума  $j$ , являющегося второй стороной взаимодействия. В общем случае можно постулировать сложную зависимость  $D(i, t)$  от обоих параметров  $i$  и  $j$ .

В случае малых приращений можно приближенно считать, что главной линейной частью такой внутренней зависимости является разность между величинами  $A(i, t)$  и  $A(j, t)$ . Такое приближение неявно, но строго логически вытекает из уже использованных выше наблюдений за феральными людьми [1].

Если при равных значениях функции величины разума для взаимодействующих индивидуумов в каких-то обстоятельствах формируется ненулевое значение приращения разума, то для ферального человека эта ситуация получалась бы при взаимодействии с самим собой. Поскольку факты говорят об отсутствии конечного эффекта подобных взаимодействий [1], то неизбежно подобное приращение всегда равно нулю. Таким образом, при малых отличиях взаимодействующих разумов, продолжая функцию по непрерывности в области нулевого значения, можно приближенно считать, что  $D(i, t) = f(A(j, t) - A(i, t))$ , причем неточность этого приближения можно также частично учесть с помощью сомножителя  $B(i, t)$ , учитывающего интенсивность взаимодействия.

**4. Обоснование метода измерения разума.** Основное, хотя и не единственное проявление разума, состоит в разумной речи. Сам по себе разговорный текст не является достаточным признаком разума, что непосредственно следует из широко известных фактов воспроизведения человеческой речи со стороны некоторых видов приручаемых птиц, в частности, попугаев и ворон. Собственно разумная содержательность текста речи в общем случае не может служить оценкой разумности объекта ее воспроизведения, поскольку и обученные речи птицы употребляют ее часто по назначению (устная просьба еды), и современные информационные системы способны по ситуации выдавать содержательные тексты. Попытки как-то универсально и формально связать содержание текстов с произвольным или хотя бы широким множеством ситуаций наталкиваются на крайне сложную проблему формального описания возможных реальных ситу-

аций и событий. Поэтому в качестве замены такого не конструируемого в настоящее время вне человеческого инструмента используем самого Homo Sapiense. В качестве первичного технического средства для оценивания величины разума будем использовать тексты, выдаваемые их носителем.

В современных условиях уже созданных средств информационных технологий не представляет принципиальной сложности фиксировать с помощью мобильных и записывающих устройств значительную часть речевых текстов, воспроизводимых человеком, в том числе в раннем возрасте.

Пусть в отдельном взаимодействии в интервале времени  $[t, t + \Delta t]$  между людьми для индивидуума  $i$ , теоретически формирующем или способном сформировать приращение  $\Delta A$  величины его разума, формируется текст  $m$ . (Для большей общности нет необходимости постулировать обязательно речевой характер этого текста, он может быть и письменным). Для каждого индивидуума  $i$  можно рассматривать множество  $M(i, t)$  всех текстов теоретически (возможно) воспринятых этим индивидуумом к моменту  $t$ , точнее услышанных и увиденных им к этому моменту.

Основной интерес представляют разумные тексты, порожденные формируемым разумом. Поэтому в качестве оценки приращения последнего будем использовать меру только таких текстов, которые произведены самостоятельно. Это требование соответствует отношению не включения в множество  $M(i, t)$ , не воспроизведение уже слышанного или прочитанного, а самостоятельно построенный текст. Самостоятельно построенный текст может быть бессмысленным, неправильным или не соотноситься явно с реальностью. Поэтому требуется эффективный и действующий критерий отбраковки таких текстов.

Для его формирования рассмотрим множество  $J_i$  людей, непосредственно взаимодействующих с рассматриваемым индивидуумом  $i$  после момента  $t_0 + \Delta t$ . Если рассматриваемый текст  $m$  не входил до этого момента в их собственные множества  $M(i, t)$ ,  $j \in J_i$ , то с этого момента оказывается входящим в их потенциальные множества текстов  $M(i, t)$ .

Далее во времени он оказывается воспроизводимым ими в их собственных взаимодействиях с другими людьми, то с содержательной стороны можно считать, что по человеческому внутреннему анализу этого текста он признан ими разумным, содержательным и прямо относящимся к некоторой моделируемой ситуации. Обозначим текст  $m$ , воспроизводимый индивидуумом  $i$  в интервале  $[t, t + \Delta t]$  как  $V(i, t)$ .

Таким образом, основанием для признания текста  $m$  самостоятельным для  $i$  является выполнение формальных условий

$$m \notin M(i, t), t < t_0, m = V(i, t_0), \quad (1)$$

$$m \notin M(j, t), j \in J_i, t < t_0, \quad (2)$$

$$m \in M(j, t), j \in J_i, t > t_0 + \Delta t, \quad (3)$$

$$m = V(j, t), t > t_0 + \Delta t.$$

Условие (1) выражает свойство новизны текста для рассматриваемого индивидуума, свойства (2) и (3) отражают новизну этого текста для субъектов

его будущих взаимодействий. Второе свойство описывает, что он был не известен им до рассматриваемого момента, а третье — фиксацию его как известного для них после взаимодействия с разумом  $i$ .

В простейшем или, скорее, начальном подходе можно мерой отдельного самостоятельного высказывания (текста) считать любую фиксированную числовую величину, например 1.

В более общем случае следует связать эту меру с числом повторного воспроизведения этого сообщения другими людьми, взаимодействующими с рассматриваемым.

**5. Проблемы использования предложенного метода.** Имеется определенная внутренняя сложность в техническом использовании уже описанного формального метода. Во множестве ситуаций под текстом взаимодействия следует понимать отдельное высказывание, построенное как законченное предложение или группу предложений с общим человеческим смыслом.

При этом выпадает довольно широкое множество высказываний, самостоятельных и разумных по существу проблемы, которое связано с конструкциями, обобщаемыми местоимением «этот» или ему подобными. Широкий и очень важный класс высказываний, связанный с построением абстракций и обобщений формируемым или развиваемым разумом, состоит в соотношении некоторого свойства, выражаемого текстом, к наблюдаемому или косвенно называемому объекту.

В свою очередь, расширение изложенного подхода на указанные общие ситуации требует от технических средств не только фиксации зрительных (в общем случае чувственных ощущений), но и эффективных средств очень глубокого распознавания их во всем многообразии деталей, что пока не видится легко осуществимым. Очень важным моментом подобного расширения является то, что он теснейшим образом связан с темой формирования понятий как классов чувственных образов и классов их текстовых производных. Эта проблема и подходы к ее решению рассматривались в работах [5, 6].

**6. Локальные факторы приращения разума.** Обратимся к рассмотрению составного сомножителя  $B(i, t)$  приращения величины разума во взаимодействии, названному интенсивностью. Учитывая, что человеческие взаимодействия оказываются явлением социальным, а не биологическим или формальным, необходимо учитывать в нем именно социальные факторы. Общеизвестно, что реальные человеческие взаимодействия существенно связаны с эмоциональной стороной человеческой личности, наличием целевых установок и волевыми возможностями взаимодействующих индивидуумов. Особенно хорошо это видно при взаимодействии с маленькими детьми, при взаимодействии обучения, в частности, в преподавательской и воспитательской деятельности, а также во взаимодействии социальной подчиненности (начальника и подчиненного).

При уже достигнутой значительной величине функции  $A(i, t)$  текущее взаимодействие с разумом той же величины может давать заметное приращение  $\Delta A$ , но при ее малой исходной величине прироста  $\Delta A$  величины разума оказывается нулевой или очень малой величиной. Действительно, феномен феральных людей указывает на нулевые приращения только для указанных малых значений, а для значительно развитого разума имеется множество

фактов, которые подтверждают накопление приращений от взаимодействий с самим собой. Такие взаимодействия характерны для профессиональных мыслителей, значительная часть внутреннего развития которых происходит от внутренних «собственных раздумий и размышлений». Реальная активная мыслительная деятельность творческих людей, как хорошо знакомо любому ученому, часто проявляется не во внешних взаимодействиях с другими людьми, а во внутренних диалогах, дискуссиях и внутренних высказываниях. Другой, и социально, и чувственно значимой деятельностью является моделирование внутри себя с помощью текстов поведения не одного активного индивидуума, а более. Наиболее простой, удобный и эффективной моделью оказывается мысленное представление внутреннего собеседника.

Для функции  $V(i,t)$  интенсивности взаимодействия можно явно выделить социальную и психологическую компоненты взаимодействия. Действительно, любое информационное взаимодействие людей обязательно имеет как индивидуально-личностную составляющую, так и социальную составляющую. Последняя обязательно присутствует, поскольку информационное взаимодействие между людьми необходимо использует информацию с общей интерпретацией, которая из-за своей общности является именно социальной.

Психологическая составляющая интенсивности взаимодействия оказывается наиболее сложной, практически наименее доступной для естественнонаучного и формализованного анализа, поскольку отражает внутренний мир человека, лишь в малой степени отражаемый доступными средствами в социальную среду и конкретные взаимодействия. В ней можно выделить психологические установки, проявляющиеся в конкретных взаимодействиях, личностные устремления и различные психологические комплексы.

Больше возможностей для внешних воздействий, представляет социальная компонента интенсивности. Социальное различие вызывает в развиваемой стороне непродизвольную эмоциональную реакцию и формирует стимул развития для менее значимой стороны, который проявляется во внутренних усилиях действий по пути к достижению такой же социальной позиции. Эта форма реакции очень хорошо заметна при взаимодействиях с известными и особенно знаменитыми деятелями культуры и искусства. Отсюда следует, что эффективным средством не только для развлекательных выступлений, но и для систематически развиваемого разума оказывается различие социальных позиций индивидуумов для сторон воздействия, которое служит эффективным средством интенсификации актов обучения.

Если же социальный статус преподавателя незначительно выше или он ниже, чем социальный статус обучаемого, то фактический эффект становится значительно ниже, несмотря на все старания, знания, навыки и умения обучающего, вплоть до практически нулевого эффекта развития разума. При этом обучаемый запоминает какие-то тексты,

правила действий, но развития его разума не происходит, он остается на том же уровне разума.

**Заключение.** Установлено, что разум, как естественнонаучное понятие, не может быть задан только качественной характеристикой, а должен описываться количественной величиной — математической мерой разума, которая допускает объективное внешнее измерение. Увеличение этой характеристики обеспечивается последовательно информационными взаимодействиями с другими разумными. В качестве них чаще всего используются взаимодействия между разумными индивидуумами. В отдельных случаях при достаточно развитом разуме возможны развивающие взаимодействия с моделируемыми внутри личности виртуальными иными личностями. Математические элементы модели позволяют количественно предсказывать и оценивать величину приращения разума для конкретных информационных взаимодействий. Из модели логически вытекает практическая необходимость высокого уровня разума для профессиональных учителей и преподавателей. Математические элементы модели научно обосновывают значение социального статуса ученых и преподавателей для реального возрастания величины разума обучаемых ими людей. Как следствие, можно прогнозировать массовое снижение средней и максимальной величины разума в социуме при систематическом снижении этого статуса. Изложенный подход может быть использован при разработке программных средств объективного оценивания величины индивидуального разума с помощью информационных технологий.

#### Библиографический список

1. Bettelheim, B. Feral Children and Autistic Children / Bruno Bettelheim // The American Journal of Sociology, Vol. 64, No. 5. (Mar., 1959), pp. 455–467.
2. Большой психологический словарь / Под ред. Б. Г. Мещерякова, В. П. Зинченко. — М.: Прайм-ЕВРОЗНАК. — 2003. — 672 с.
3. Соловьёв, В. С. Разум / В. С. Соловьёв // Энциклопедический словарь Брокгауза и Ефрона. В 86 т. (82 т. и 4 доп.). Т. XXVI. — СПб., 1899. — С. 204–205.
4. Флоренсов, А. Н. Ноосферные аспекты кибернетики и проблема воспроизводства разума / А. Н. Флоренсов // Вестник Тамбовского ун-та. Сер. Естественные и технические науки. — 2013. — Т. 18, вып. 3. — С. 1084–1087.
5. Флоренсов, А. Н. Построение абстрактного пространства для семантической теории информации / А. Н. Флоренсов // Доклады Сибирского отделения Академии наук высшей школы. — 2000. — № 2. — С. 94–101.
6. Флоренсов, А. Н. Динамические аспекты информационной компоненты сложных систем: моногр. / А. Н. Флоренсов. — Омск: Изд-во ОмГТУ, 2007. — 344 с.

**ФЛОРЕНСОВ Александр Николаевич**, кандидат технических наук, доцент (Россия), доцент кафедры «Информатика и вычислительная техника». Адрес для переписки: florensov@yandex.ru

Статья поступила в редакцию 30.11.2015 г.

© А. Н. Флоренсов

## АДАПТИВНОСТЬ В ПРОЦЕССАХ ИНФОРМАЦИОННОЙ ПОДДЕРЖКИ ПЕРСОНАЛА ПРОМЫШЛЕННОГО ПРЕДПРИЯТИЯ

**В данной статье рассмотрены особенности организации информационной поддержки сотрудников промышленного предприятия. На основе действующего производства произведен анализ структуры единого информационного пространства. Представлено формализованное описание, на основе которого разработан механизм адаптации информационной выдачи, а также рассмотрен опыт его внедрения.**

**Ключевые слова:** данные, релевантность, доступность, производственный процесс, единое информационное пространство, адаптивность.

**Введение.** Современные цифровые библиотеки и связанные с ними системы, к которым, несомненно, относятся индустриальные репозитории, сталкиваются с необходимостью взаимодействовать с разнообразно структурированными информационными единицами [1].

Они представлены в виде разнообразных цифровых форм и имеют свойство быть произведенными в среде цифрового контента. Мы переходим в эпоху, когда материальные носители отходят на второй план и сам контент уже производится в виртуальной среде, а не оцифровывается из материальной. Примером могут служить результаты интеллектуальных изысканий, институциональных или индустриальных архивов, нематериального производственного наследия (конкретный уникальный опыт предприятия) и даже контент, создаваемый пользователями, такой как примечания и рекомендации.

Такие информационные единицы, как наборы эксплуатационной документации, могут происходить из разнородных источников, что довольно часто происходит на практике, в том числе хранилищ XML или обычных бизнес-решений для баз данных [2]. Постоянно увеличивающиеся возможности передачи данных различными протоколами, обеспечиваемые за счет расширения сети и повышения скорости передачи, создают глобальную проблему по организации взаимосвязей и подключения таких гетерогенных источников и интеграции их в механизмы формирования информационной выдачи по запросам, поступающим от специалистов.

Фокусируясь на процессах информационной поддержки как в рамках одного промышленного предприятия, так и целого комплекса, можно утверждать, что любые подходы по модернизации данных процессов, которые включают в себя ручное выполнение сложных задач, скорее всего, будут неэффективными на практике [3].

Достаточно ясно, что для эффективной информационной поддержки персонала промышленного предприятия необходимо адаптировать информа-

ционную выдачу к конкретным задачам в автоматизированном режиме. Для этого требуется механизм, способный производить множество операций по открытию информации, организации доступа, концептуализации опций, поддерживаемых привлеченными системами [4]. Таким образом, необходимо на практике расширить различные аспекты управления информацией участвующих систем [5].

**Теоретический анализ.** В рамках производства как предметной области мы должны произвести описание самой области и её объектов, чтобы обеспечить возможность однозначно описать объект, например, изделие в контексте текущего состояния среды с учетом задач текущего агента и адаптировать информационную выдачу под его нужды [6–7].

Это предполагает наличие различных типов маркеров-инициализаторов построения информационной выдачи и агентов-реципиентов, а также механизма адаптации. Такая подсистема должна обеспечивать следующее: поддержку механизмов обнаружения новой информации, организацию доступа к новым видам информационных источников, дополнение системы новыми сценариями использования контента.

Прибегнем к теоретико-множественному описанию объектов предметной области [8]. В общем случае элементы предметной области для обеспечения возможности информационной поддержки в адаптивном режиме необходимо формально представить следующим кортежем информации:

$$(\text{State}, \text{Obj}, \text{Prop}, \text{TypeObj}, \text{TypeProp}, \text{InfNode}, \text{TypeInfNode}, \text{InfCell}, \text{TypeInfCell}), \quad (1)$$

где

- *State* — множество состояний предметной области;
- *Obj* — множество объектов предметной области;
- *Prop* — множество свойств предметной области;



- *TypeObj* — множество типов объектов предметной области;
- *TypeProp* — множество типов свойств предметной области;
- *InfNode* — множество информационных нод;
- *TypeInfNode* — множество типов информационных нод;
- *InfCell* — множество информационных ячеек;
- *TypeInfCell* — множество типов информационных ячеек.

Рассмотрим подробнее.

$$\begin{aligned} State_i \in State, State_i = \\ = (Time, Mark, Params, Label), \end{aligned} \quad (2)$$

где  $i$  — уникальный внутренний идентификатор состояния,  $i \in N$ , где  $N$  — множество натуральных чисел;

$State_i$  — элемент из множества состояний предметной области, который указывает текущее состояние.

Используется для характеристики эффективности текущего состояния.

Аналитика состояний и ведение контрольных точек необходимы для эффективной эволюции ЕИП;

$Time_i \in Tim, Time_i$  — элемент из множества временных меток состояния предметной области;

$Mark_i \in Mark, Mark_i$  — элемент из множества оценок состояния предметной области;

$Params_i \in Params, Params_i$  — элемент из множества «слепков» значений всех основных параметров предметной области;

$Label_i \in Label, Label_i$  — уникальный внешний символьный идентификатор состояния предметной области, задается пользователем системы.

Множество информационных объектов предметной области определяется следующим образом:

$$Obj_i \in Obj, Obj_i = (TypeObj, Label), \quad (3)$$

где  $i$  — уникальный внутренний идентификатор объекта,  $i \in N$ , где  $N$  — множество натуральных чисел;

$TypeObj_i \in TypeObj, TypeObj_i$  — элемент из множества типов объектов предметной области, указывает тип объекта;

$Label_i \in Label, Label_i$  — уникальный внешний символьный идентификатор информационного объекта, задается пользователем системы.

Множество свойств предметной области определяется следующим кортежем информации:

$$Prop_i \in Prop, Prop_i = (TypeProp, Label), \quad (4)$$

где  $i$  — уникальный внутренний идентификатор свойства предметной области,  $i \in N$ , где  $N$  — множество натуральных чисел;

$TypeProp_i \in TypeProp, TypeProp_i$  — элемент из множества типов свойств предметной области, указывает тип свойства;

$Label_i$  — уникальный внешний символьный идентификатор свойства предметной области, задается пользователем системы.

В предлагаемом методе необходимым является введение следующих архетипов для объектов:

- реальные объекты;
- *IniMarker* — инициализирующий маркер быстрого доступа;
- *IniMarkerHolder* — обладатель инициализирующего маркера быстрого доступа;

— *Tool* — инструмент или средство производства (станок, инструмент);

— *Component* — компоненты для производства (детали изделия);

— *Agent* — агент (сотрудники предприятия);

— виртуальные объекты;

— *Doc* — общий документный тип;

— *Manual* — руководство;

— *Advice* — практические рекомендации;

— *Note* — заметки;

— *Notify* — уведомления и прочая приоритетная информация, которая имеет глобальный характер.

Таким образом, объект может соответствовать нескольким вложенным типам, например, «Tool» и «IniMarkerHolder».

Следовательно, множество типов объектов предметной области определяется следующим образом:

$$\begin{aligned} TypeObj \in (IniMarker, IniMarkerHolder, \\ Tool, Component, Agent, Doc, \\ Manual, Advice, Note, Notify), \end{aligned} \quad (5)$$

Множество типов свойств предметной области определяется следующим образом:

$$\begin{aligned} TypeProp \in (counter, string, \\ number, date, cost, boolean, link), \end{aligned} \quad (6)$$

где

— *counter* — «счетчик» — целые числа, которые задаются автоматически при вводе записей. Эти числа не могут быть изменены пользователем;

— *string* — «текстовый» тип свойств предметной области — тексты, содержащие до 255 символов;

— *number* — «числовой» тип свойств предметной области — числа;

— *date* — тип свойств предметной области «дата/время» — дата или время;

— *cost* — «денежный» тип свойств предметной области — числа в денежном формате;

— *boolean* — «логический» тип свойств предметной области — значения Истина (Да) или Ложь (Нет);

— *link* — тип свойств предметной области «гиперссылка» — ссылки на информационный ресурс.

Каждому сотруднику на производстве задается уникальный агент [9, 10]. Множество агентов определяется следующим кортежем информации:

$$\begin{aligned} Agent_i \in Agent, Agent_i = \\ = (TypeAgent, RatingRel, RatingSp, Label), \end{aligned} \quad (7)$$

где  $i$  — уникальный внутренний идентификатор свойства предметной области,  $i \in N$ , где  $N$  — множество натуральных чисел;

$TypeAgent_i \in TypeAgent$  — элемент из множества типов агентов, указывает тип агента;

$RatingRel_i \in RatingRel$  — элемент из множества уровня релевантности получаемых данных, указывается индивидуальное значение для каждого агента;

$RatingSp_i \in RatingSp$  — элемент из множества уровня скорости удовлетворения информационных потребностей, указывается индивидуальное значение для каждого агента;

$Label_i$  — уникальный внешний символьный идентификатор агента, задается пользователем системы.

В предлагаемом методе для обеспечения работы механизма по повышению релевантности информации необходимо введение такого комплексного параметра, как тип агента, «TypeAgent», который позволяет нам определять следующее:

- привилегии;
- приоритет в обслуживании информационных потребностей;
- уровень доступа к данным;
- функции агента;
- типовые информационные потребности;
- индивидуальные информационные потребности.

Множество инициализирующих маркеров быстрого доступа определяется следующим кортежем информации:

$$\begin{aligned} IniMarker_i \in IniMarker, IniMarker_i = \\ = (TypeIniMarker, DataM, Label, Obj), \end{aligned} \quad (8)$$

где  $i$  — уникальный внутренний идентификатор свойства предметной области,  $i \in N$ , где  $N$  — множество натуральных чисел;

$TypeIniMarker_i$  — элемент из множества типов инициализирующих маркеров быстрого доступа, указывает тип маркера (оффлайн, онлайн, рабочий режим, экстренная ситуация);

$DataM_i$  — элемент из множества данных внутри инициализирующих маркеров быстрого доступа;

$Label_i$  — уникальный внешний символичный идентификатор маркера, задается пользователем системы;

$Obj_i$  — элемент из множества объектов, он показывает, к какому объекту-носителю относится каждый маркер.

Множество информационных ячеек определяется следующим кортежем информации:

$$\begin{aligned} InfCell_i \in InfCell, InfCell_i = \\ = (TypeInfCell, DataC, Label, Obj, Prop_i), \end{aligned} \quad (9)$$

где  $i$  — уникальный внутренний идентификатор свойства предметной области,  $i \in N$ , где  $N$  — множество натуральных чисел;

$TypeInfCell_i$  — элемент из множества типов информационных ячеек, это комплексный параметр, который позволяет нам определять следующее:

- необходимые привилегии для считывания (отсылка к типам агентов);
- приоритет в обслуживании информационных потребностей;
- уровень доступа к данным;
- указывается градация степени релевантности для типа агента;
- характер данных (текстовый, графический, медиа, смешанный и т.п.);

$DataC_i$  — элемент из множества данных внутри информационных ячеек;

$Label_i$  — уникальный внешний символичный идентификатор ячейки, задается пользователем системы;

$Obj_i$  — элемент из множества объектов, отражает, какой объект описывается ячейкой;

$Prop_i$  — элемент из множества свойств предметной области, отражает, какое свойство описывается ячейкой.

Проиллюстрируем в общем виде теоретико-множественное описание на основе типовых операций с изделием, а именно его сборкой и разборкой.

В предыдущих статьях из этого цикла была предложена математическая модель, в основе которой лежит понятие состава продукта:

$$PR_{(product)} = \{ detail_1, \dots, detail_n \}, \quad (10)$$

где  $i = 1 \dots n$  — определяет, сколько тех или иных типов деталей в продукте;

$detail_i$  — коэффициент входимости для детали  $i$ .

Под продуктом понимается композиция деталей. Также предлагаемая модель учитывает, что детали могут быть заменяемы «кроссами», а само изделие может выступать как деталь более сложного продукта или иметь несколько версий/вариантов исполнения.

В общем случае сценарий как последовательность действий в рамках определенных временных интервалов над множеством деталей изделия можно представить следующим кортежем информации:

$$(S, P, D, A, V, T, Al, Ps, Pp, Pd, Ad), \quad (11)$$

где

- $S$  — множество сценариев;
- $P$  — множество изделий;
- $D$  — множество деталей;
- $A$  — множество действий;
- $V$  — множество версий объектов предметной области;
- $T$  — множество временных интервалов;
- $Al$  — множество альтернативных сценариев;
- $Ps$  — множество свойств сценария;
- $Pp$  — множество свойств изделия;
- $Pd$  — множество свойств детали;
- $Ad$  — множество свойств действия.

При описании сценария вводится дополнительный параметр — «доступность». Таким образом, мы составляем алгоритм, описывающий состав действий над объектами, последовательность этих действий и эффект при каждом шаге. Описание производится в следующем порядке: действие, объект действия, доступность новых объектов.

Благодаря такому представлению сценария, мы можем находить альтернативные пути достижения поставленных на производстве целей, использовать возможность адаптивного использования стандартных сценариев сборки и разборки (например, в задачах, когда максимально быстро необходимо получить доступ к определенной группе деталей изделия).

**Методика.** При проведении данного исследования использовались общая теория систем, теория управления, системный анализ, теория множеств.

Произведена формализация единого информационного пространства промышленного предприятия. Введен ряд ключевых понятий и классификаций.

Выполнен анализ процессов взаимодействия сотрудников предприятия с корпоративным информационным хранилищем.

Было проведено наблюдение за работой специалистов на производстве с имеющимися системами информационной поддержки, в частности, работа с эксплуатационной документацией. По результатам наблюдения было составлено описание соответствующих процедур, введен ряд критериев, к примеру, «Отказы от информационной выдачи», а также подготовлен практический эксперимент.

В рамках практического эксперимента была подтверждена гипотеза о повышении эффективности

информационной поддержки путем введения механизмов адаптивной выдачи.

Произведено сравнение в рамках работы специалистов как с классическими системами по методу прямого запроса, так и с разработанной авторской системой.

**Результаты.** Разработано и испытано программное обеспечение «QR-Docs» для быстрого доступа к эксплуатационной документации и проведено его тестирование в различных режимах работы.

Основные результаты работы нашли применение в промышленности в области машиностроения, в частности, на базе ООО «Вологодские машины» (г. Вологда) и ОАО «Северный Коммунар» (г. Вологда).

#### Библиографический список

1. Сценарий и механизмы создания ЕИП ракетно-космической промышленности. Электронный архив технической документации как основа ЕИП / А. М. Воробьев [и др.] // CADmaster. — 2010. — № 5. — С. 26–51.
2. Бахтенко, Е. А. Модель процесса разборки сложной продукции с учетом особенностей ее состава и структуры / Е. А. Бахтенко // Молодой ученый. — 2014. — № 9. — С. 112–116.
3. Ершова, Т. Б. Организационные аспекты создания единого информационного пространства предприятия [Текст] / Т. Б. Ершова // Транспортное дело России. — 2009. — № 2. — С. 56–57.
4. Измайлов, И. А. Проектирование многоагентных систем в управлении сетевыми ресурсами предприятия / И. А. Измайлов, Е. А. Бахтенко, А. А. Суконщиков // Автоматизация и энергосбережение машиностроительного и металлургического производств, технология и надежность машин, приборов и оборудования : материалы 7-й Междунар. науч.-техн. конф. — Вологда : ВоГТУ, 2012. — С. 66–70.
5. Основные направления и результаты работ по применению CALS-технологий для повышения качества и конкурен-

тоспособности военной продукции [Текст] / А. Г. Кабанов, А. Н. Давыдов, В. В. Барабанов, Е. В. Судов // Информационные технологии в проектировании и производстве. — М. : ГУП «ВИМИ». — 2000. — № 2. — С. 3–6.

6. Концепция формирования и развития единого информационного пространства России и соответствующих государственных информационных ресурсов [Текст] // Информационное общество. — М. : Роспечать, 1995. — № 4. — С. 22–24.

7. Концепция развития CALS-технологий в промышленности России [Текст] / Е. В. Судов [и др.]. — М. : НИЦ CALS-технологий «Прикладная логистика», 2002. — 36 с.

8. Суконщиков, А. А. Спирально-адаптивная схема жизненного цикла информационной системы / А. А. Суконщиков, Д. А. Малышев // Автоматизация и энергосбережение машиностроительного и металлургического производств, технология и надежность машин, приборов и оборудования : материалы 6-й Междунар. науч.-техн. конф. — Вологда : ВТУ, 2010. — Т. 1. — С. 187–189.

9. Суконщиков, А. А. Масштабирование параллельных систем / А. А. Суконщиков, Д. А. Яковлев // Информационные технологии в проектировании и производстве. — М. : ВИМИ. — 2011. — № 2. — С. 66–69.

10. Суконщиков, А. А. Обобщенная модель системы ситуационного интеллектуально-агентного моделирования / А. А. Суконщиков, Д. А. Яковлев // Информационно-управляющие системы. — СПб. : Политехника. — 2010. — Т. 45, № 2. — С. 9–14.

**БАХТЕНКО Евгений Андреевич**, аспирант кафедры автоматизации и вычислительной техники.

Адрес для переписки: evgars@gmail.com

**СУКОНЩИКОВ Алексей Александрович**, кандидат технических наук, доцент (Россия), заведующий кафедрой автоматизации и вычислительной техники.

Адрес для переписки: avt@vstu.edu.ru

Статья поступила в редакцию 28.04.2015 г.

© Е. А. Бахтенко, А. А. Суконщиков

## Книжная полка

004/Н64

**Никонов, А. В. Заключительная аттестационная работа бакалавра, магистра и специалиста : [Электронный ресурс] : учеб. пособие / А. В. Никонов, В. Н. Цыганенко. — Омск : Изд-во ОмГТУ, 2015. — 1 о=эл. опт. диск (CD-ROM).**

В учебном пособии рассматривается структура многоуровневой системы подготовки специалистов с высшим техническим образованием в РФ, излагается механизм итоговой государственной аттестации выпускников ОмГТУ по направлению 230100 «Информатика и вычислительная техника» и специальности 230102 «Автоматизированные системы обработки информации и управления». Приводятся нормативные требования и рекомендации по выполнению, оформлению и защите выпускных квалификационных работ бакалавров, магистров, дипломированных специалистов. Учебное пособие рекомендуется студентам выпускных курсов направления 230100 и специальности 230102. Может быть полезно научным руководителям и рецензентам выпускных квалификационных работ.

004.5/Б28

**Батенькина, О. В. Юзабилити информационных систем : учеб. пособие для вузов по направлению подгот. бакалавра 09.03.02 «Информационные системы и технологии» / О. В. Батенькина, О. Н. Ткаченко. — Омск : Изд-во ОмГТУ, 2015. — 143 с.**

Содержит теоретические и прикладные вопросы юзабилити-тестирования программных интерфейсов. Описаны конкретные процедуры и программно-методические средства для юзабилити-тестирования интерфейсов информационных систем. Даны рекомендации в области юзабилити для проектировщиков программных интерфейсов. Предназначено для студентов высших учебных заведений, обучающихся по направлению подготовки бакалавра 09.03.02 «Информационные системы и технологии».

## МЕТОДОЛОГИЯ РЕИНЖИНИРИНГА БИЗНЕС-ПРОЦЕССОВ И ТИПОВЫЕ ПРИЕМЫ ЕЁ ПРИМЕНЕНИЯ

Нестабильная экономическая ситуация в России и в мире требует от организаций быстрого реагирования на изменения, высоких конкурентных преимуществ и сохранения устойчивости бизнеса. Реинжиниринг является самым радикальным методом совершенствования бизнеса. Кроме того, несмотря на наличие достаточно большого количества публикаций и книг о данном методе, около половины проектов, основанных на этом подходе, заканчивается неудачей. Причиной этого является, главным образом, неверное представление о реинжиниринге. В рамках данной работы рассмотрены особенности методологии реинжиниринга бизнес-процессов. В ее основе лежит процессный подход к управлению предприятием и понятие бизнес-процессов. Методология также рассмотрена с точки зрения управления проектами, выделены основные способы и приемы для проведения реинжиниринга. Для достижения задач реинжиниринга используются информационные технологии, позволяющие моделировать бизнес-процессы.

**Ключевые слова:** реинжиниринг, процессный подход, бизнес-процессы, проект.

Центральным понятием методологии реинжиниринга является бизнес-процесс. Бизнес-процессы существуют внутри каждой организации и представляют собой процессы из последовательности операций на предприятии, которые направлены на преобразование входных информационно-материальных потоков с целью получения результатов, представляющих ценность для клиента [1].

Применение в организации системы процессов наряду с их идентификацией и взаимодействием, а также управление процессами, направленное на получение желаемого результата, могут быть определены как «процессный подход». Преимущество процессного подхода состоит в непрерывности управления, которое он обеспечивает на стыке отдельных процессов в рамках их системы, а также при их комбинации и взаимодействии [2].

В основе процессного подхода к управлению организацией лежит выделение в организации бизнес-процессов и управление этими бизнес-процессами. Управление бизнес-процессами — концепция процессного управления организацией, рассматривающая бизнес-процессы как особые ресурсы предприятия, непрерывно адаптируемые к постоянным изменениям. Концепция полагается на такие принципы, как понятность и видимость бизнес-процессов в организации за счёт их моделирования с использованием формальных нотаций и программного обеспечения [3].

Управление бизнес-процессами организации предполагает их постоянное улучшение, поэтому важнейшими инструментами процессного управления являются методы совершенствования бизнес-процессов.

Существует четыре основных подхода к совершенствованию бизнес-процессов: бенчмаркинг, перепроектирование, реинжиниринг, методика быстрого анализа решения (FAST). Реинжиниринг является самым радикальным из всех четырех подходов к улучшению бизнес-процессов. Подход обеспечивает свежий взгляд на цели процесса и структуру организации.

Понятие «реинжиниринг бизнес-процессов» (Business process reengineering, BPR) возникло в 1990 г. в США (авторы — М. Хаммер и Д. Чампи). Под реинжинирингом бизнес-процессов (РБП) понимают фундаментальное переосмысление и радикальное перепроектирование бизнес-процессов для достижения существенных улучшений в таких ключевых для современного бизнеса показателях результативности, как затраты, качество, уровень обслуживания и оперативность. Задача реинжиниринга бизнес-процессов — найти совершенно новый способ реконструирования существующего бизнеса, с использованием новых технических достижений для лучшего обслуживания клиентов [4, 5].

Существует два способа применения РБП:

1. Систематический реинжиниринг — текущий процесс понят, документирован и проанализирован для систематического создания новых и лучших процессов. Этот вид реинжиниринга проводится, опираясь на правила ESIA (уничтожить, упростить, объединить, автоматизировать).

2. Реинжиниринг с чистого листа — существующий процесс полностью утилизируется, а новый процесс создается с нуля путем фундаментального переосмысления существующего. Данный способ снижает риск повторения старых ошибок. Однако



пренебрежение ранее применявшимся процессом может привести к игнорированию знаний и опыта, накопленных в организации.

Существуют типовые способы и приемы, применяемые при перепроектировании бизнес-процессов в ходе реинжиниринга.

1. Горизонтальное сжатие процесса — объединение нескольких рабочих процедур в одну работу. Прием позволяет уменьшить число ошибок и сократить количество работников, занятых устранением их последствий, улучшить управляемость производством за счет четко распределенной ответственности между сотрудниками.

2. Вертикальное сжатие процессов — предоставление исполнителям права принимать самостоятельные решения. Увеличение роли каждого сотрудника в работе компании уменьшает время задержек, снижает стоимость продукции и услуг, ускоряет реакцию на запросы клиентов и т.д.

3. Выполнение шагов каждого процесса в естественном порядке — использование, наряду с линейным упорядочиванием работ, параллельное их проведение, приводит к сокращению времени и затрат.

4. Исполнение процессов в различных вариантах — процесс начинается с проверочной процедуры, цель которой определить, какой вариант его реализации наиболее подходит в заданной производственной ситуации.

5. Выполнение каждой работы там, где это наиболее целесообразно, — распределение работы между подразделениями, устранение излишней интеграции.

6. Уменьшение числа проверок — минимизация числа согласований путем сокращения внешних точек контакта и, как следствие, стирания граней между функциональными подразделениями.

7. «Уполномоченный» менеджер обеспечивает единую точку контакта — если процесс сложен и интеграция невозможна, то «уполномоченный» менеджер отвечает на вопросы заказчика и решает его проблемы, имеет доступ ко всем ИС, используемым в процессе, и ко всем исполнителям.

8. Преобладание смешанного централизованно-децентрализованного подхода — применение современных информационных технологий дает возможность компании действовать на уровне подразделений автономно, сохраняя за ними возможность использования централизованных данных [1].

Как и любой другой метод совершенствования бизнес-процессов на предприятии, реинжиниринг имеет свои преимущества и недостатки. Преимуществом методологии связаны в основном с совершенствованием структуры управления предприятием (переход к проектной структуре), снижением затрат и повышением конкурентоспособности организации. Неопределенность результатов, отсутствие значительного опыта в использовании данного метода, а также длительные сроки можно выделить в качестве главных недостатков реинжиниринга.

Методология реинжиниринга бизнес-процессов применяется на предприятиях в качестве проекта. Таким образом, проект по РБП — это целенаправленная деятельность, результатами которой являются преобразования набора бизнес-процессов организации, необходимые для достижения бизнес-целей организации. Проект по РБП имеет ограничения по срокам и бюджету и выполняется командой участников, специально организованной на время проекта и возглавляемой руководителем проекта.

Проект РБП можно разделить на фазы планирования, выработки решений, внедрения. Решения, вырабатываемые в РБП, могут радикально уходить от существующего процесса. Поэтому вторую фазу можно разбить еще на две: первая из них связана с фактической генерацией предложений по изменениям, а вторая — с оценкой того, как их можно внедрить.

Следовательно, в процессе проведения РБП можно выделить следующие этапы:

1. Планирование — определяется проект РБП, формируется команда проекта, определяются цели проекта. Задачи, решаемые на этом этапе:

- выбор процесса для совершенствования с помощью РБП и определение его масштаба;

- оценка возможностей достижения улучшений и установление целей;

- создание команды для работы в проекте;

- разработка плана проекта РБП. Детальность плана зависит от масштаба и сложности.

Основные моменты, которые должны быть отражены в плане:

- действия проекта;

- кто их будет выполнять;

- когда их следует выполнить;

- какие ресурсы нужны и какими ресурсами вы располагаете для проекта;

- ожидаемые результаты проекта.

2. Реинжиниринг. Основная цель этой фазы — реинжиниринг процесса, направленный на улучшения. Основные шаги на данном этапе:

- документирование существующего процесса;

- реинжиниринг процесса. Выбор альтернативы проведения реинжиниринга (систематический или «с чистого листа») во многом зависит от процесса;

- выработка рекомендаций по улучшению. Полученный перечень рекомендуемых изменений составит базу для работы в следующей фазе — фазе принятия решений о реализации выработанных содержательных предложений.

3. Преобразование. Основная цель этой фазы — построение базиса для эффективного и успешного внедрения нового процесса. Главные задачи этой фазы:

- оценка перемен, требуемых для внедрения нового процесса;

- планирование требуемых инвестиций, обучения персонала, закупок и т.д.

- создание благоприятного климата для перемен;

- планирование внедрения.

4. Внедрение. Основные шаги фазы внедрения таковы:

- определение набора целей для совершенствования;

- реализация плана внедрения;

5. Мониторинг прогресса внедрения и устранение любых отклонений [4, 6].

Применение стандартов управления проектами в процессе реинжиниринга позволит расширить содержание методики и привести в нее новые элементы, такие как анализ рисков, управление качеством и др. [2].

При проведении реинжиниринга организация претерпевает ряд существенных изменений.

1. Переход от функциональных подразделений к командам процессов — люди объединяются в группы, которые совместно выполняют процесс. Выделяют три типа команд по следующим признакам:

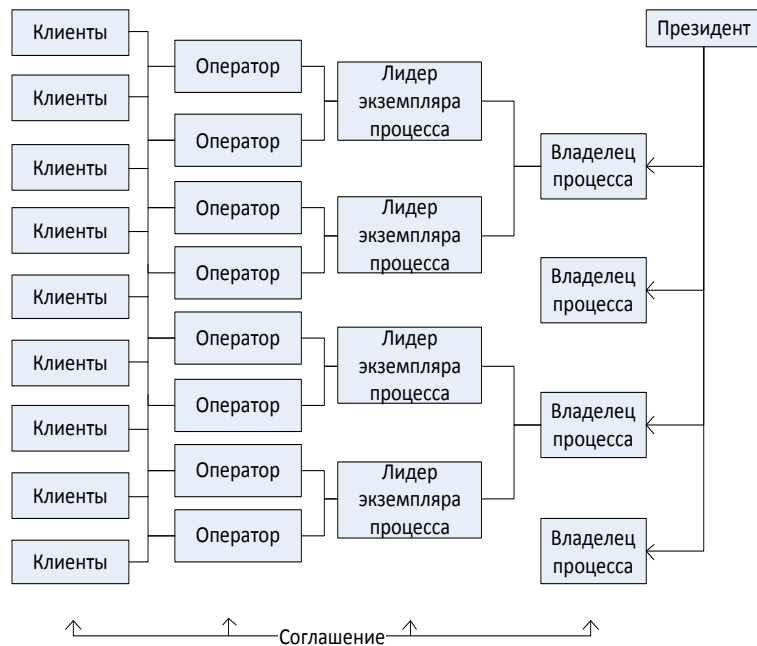


Рис. 1. Структура компании после проведения реинжиниринга

— команда, объединяющая некоторое число совместно работающих людей с различными специальностями, выполняющих рутинную работу; члены команды объединяются на длительное время;

— «виртуальная» команда, объединяющая людей для решения эпизодической и, как правило, очень сложной задачи; после достижения поставленной цели команда расформировывается и ее члены переходят в другие команды и проекты; один и тот же работник может быть членом нескольких «виртуальных» команд;

— команда, состоящая из одного человека, оснащенного необходимыми информационными и экспертными системами.

2. Работа исполнителя изменяется от простой к многоплановой — каждый член команды ответственен за весь процесс, что требует от него понимания этого процесса в целом и умения выполнять несколько заданий. В результате реинжиниринга устраняются проверки, согласования и ожидания, вызванные преодолением границ между подразделениями компании; члены команды фокусируют свои усилия на потребностях клиентов. Работу нескольких экспертов может выполнять один подготовленный специалист.

3. Изменяются требования к работникам — РБП стимулирует переход от курсов разового обучения сотрудников компании к их непрерывному образованию и самообразованию. Работники становятся ответственными за весь процесс и должны уметь выполнять любое задание.

4. Изменяются оценка эффективности и оплата труда — РБП стимулирует переход от оценки деятельности сотрудника к оценке результатов его труда, в новой компании эффективность работы сотрудников (команд) соизмерима, поскольку они выполняют законченную работу. Оплата в соответствии с полученным результатом.

5. Изменяется критерий продвижения в должности — в новой компании продвижение происходит лишь при наличии у сотрудника бесспорных способностей успешно работать на новом месте.

6. Изменяется цель исполнителя — РБП поощряет переход от удовлетворения потребностей начальства к удовлетворению потребностей клиента; например, основная часть премии менеджеров должна зависеть от степени удовлетворения ими запросов клиентов, а не от решений начальства.

7. Изменяются функции менеджеров — функции становятся тренерскими, поскольку после проведения реинжиниринга бизнес-процессов каждая команда полностью отвечает за свой процесс и управляющие воздействия менеджеров «сверху» ей не нужны.

8. Изменяются административные функции — от секретарских к лидирующим, когда администрация компании исполняет роль лидера, способствующего процветанию коллектива, укреплению убеждений и ценностей исполнителей; обеспечивающего мотивацию членов команд и заинтересованность в результатах труда.

Также при проведении реинжиниринга бизнес-процессов меняется организационная структура компании. Принцип организации новой структуры соответствует пирамиде, лежащей на боку, то есть клиенты находятся не внизу, а являются равноправными по отношению к компании (рис. 1).

Исполнителями, взаимодействующими с клиентами, являются операторы процессов, а менеджерами среднего звена — лидеры (руководители) экземпляров процессов [1].

Как было отмечено ранее, для проведения реинжиниринга бизнес-процессов необходимо задокументировать и смоделировать существующие процессы. Для осуществления этой деятельности используются программные продукты для моделирования бизнес-процессов. Их основное назначение — визуализация технологических и управленческих цепочек действий, событий и функций. Чаще всего такие продукты задействованы в проектах по реорганизации и реинжинирингу бизнес-процессов. Наиболее распространенные программные продукты этой группы:

— ARIS Toolset;

- BPWin;
- Business Studio.

Программные продукты такого типа позволяют проводить анализ, документирование и улучшение бизнес-процессов, моделировать действия в процессах, определять их порядок и необходимые ресурсы. С их помощью можно строить диаграммы различного уровня и назначения. Различные диаграммы, созданные в таких программных продуктах отражают общее представление процесса, его взаимосвязь с внешней средой, необходимые ресурсы, полученные результаты, взаимосвязи между процессами, элементами системы, определяют последовательность работ, позволяют рассчитать затраты, стоимость производства, идентифицировать самые дорогостоящие процессы.

#### Выводы:

- процессный подход рассматривает деятельность предприятия как совокупность процессов;
- под бизнес-процессом понимается последовательность связанных между собой операций, направленных на создание продукта, имеющего ценность для потребителя;
- реинжиниринг бизнес-процессов — это фундаментальное переосмысление и радикальное перепроектирование бизнес-процессов для достижения существенных улучшений в ключевых показателях результативности. Выделяют реинжиниринг «с чистого листа» и систематический реинжиниринг;
- методику реинжиниринга необходимо рассматривать не только с позиций концепции процессного подхода к управлению организацией, но и с позиции стандартов управления проектами;
- для достижения задач реинжиниринга используются информационные технологии, с помощью которых можно проводить анализ, документирование и улучшение бизнес-процессов.

## Книжная полка

#### 004.7/Г27

**Гегечкори, Е. Т. Информационные системы в производстве : [Электронный ресурс] : учеб. пособие / Е. Т. Гегечкори, О. Б. Малков. — Омск : Изд-во ОмГТУ, 2015. — 1 о=эл. опт. диск (CD-ROM).**

Излагаются основные концепции управления, широко используемые на промышленных предприятиях различного профиля во всем мире. Рассматриваются эволюция систем управления — от самых простых к более сложным (MRP-системы), новые, только получающие широкое распространение компьютерные системы (APS-системы), теория ограничений и теория «Точно вовремя», некоторые практические аспекты использования корпоративных компьютерных систем (ERP-систем), а также место ERP-систем в информационном пространстве предприятия и возможности интеграции ERP-систем с другими системами. Учебное пособие предназначено для студентов заочной формы обучения специальности 080801 «Прикладная информатика (в экономике)», а также для сотрудников отделов информационных технологий компаний, специалистов по управлению — всех тех, кто будет выбирать, внедрять и использовать эти системы.

#### 004.9/Б28

**Батенькина, О. В. Технологии анимации : учеб. пособие для вузов по направлению подгот. бакалавра 09.03.02 «Информационные системы и технологии» / О. В. Батенькина. — Омск : Изд-во ОмГТУ, 2015. — 115 с.**

Содержит материалы об основных техниках анимации, технологиях создания различных видов компьютерной анимации; включает описание этапов разработки проектов в области профессиональной деятельности средствами компьютерной графики и анимации. Предназначено для студентов высших учебных заведений, обучающихся по направлению подготовки бакалавра 09.03.02 «Информационные системы и технологии».

## Библиографический список

1. Чаадаев, В. К. Бизнес-процессы в компаниях связи / В. К. Чаадаев. — М. : Эко-Трендз, 2004. — 176 с.
2. Горюнов, Е. В. Управление проектом по реинжинирингу бизнес-процессов [Электронный ресурс] / Е. В. Горюнов // Менеджмент в России и за рубежом. — 2008. — Режим доступа : <http://www.mevriz.ru/articles/2008/3/5155/>. — Загл. с экрана (дата обращения: 14.09.2015).
3. Елиферов, В. Г. Бизнес-процессы. Регламентация и управление : учеб. / В. Г. Елиферов, В. В. Репин. — М. : ИНФРА-М, 2005. — 319 с.
4. Реинжиниринг бизнес-процессов. Полный курс МВА : учеб. / Н. М. Абдикеев [и др.] ; под ред. Н. М. Абдикеева, Т. П. Данько. — М. : ЭКСМО, 2005. — 592 с.
5. Хаммер, М. Реинжиниринг корпорации. Манифест революции в бизнесе = Reengineering the corporation: A manifesto for business revolution / М. Хаммер, Д. Чампи. — N.Y. : Harper-Collins, 2003. — 272 с.
6. Андерсен, Б. Бизнес-процессы. Инструменты совершенствования / Б. Андерсен ; пер. с англ. С. В. Ариничева. — М. : РИА «Стандарты и качество», 2003. — 272 с.

**ЗЛАТКИНА Алла Александровна**, магистрант гр. ПИМ-151 факультета элитного образования и магистратуры Омского государственного технического университета (ОмГТУ); системный аналитик ООО «ЛЮКСОФТ», г. Омск.

Адрес для переписки: [azlatkina93@mail.ru](mailto:azlatkina93@mail.ru)

**ГЕГЕЧКОРИ Евгений Тратович**, кандидат технических наук, доцент (Россия), доцент кафедры «Математические методы и информационные технологии в экономике» ОмГТУ.

Адрес для переписки: [getnbn@inbox.ru](mailto:getnbn@inbox.ru)

Статья поступила в редакцию 22.12.2015 г.

© А. А. Златкина, Е. Т. Гегечкори