Intellectual Technologies on Transport
No 2

Интеллектуальные технологии на транспорте
№ 2

Санкт-Петербург
St. Petersburg
2019

Интеллектуальные технологии на транспорте
№ 2, 2019

Сетевой электронный научный журнал, свободно распространяемый через Интернет.
Публикуются статьи на русском и английском языках с результатами исследований и практических достижений в области интеллектуальных технологий и сопутствующих им научных исследований.

Журнал основан в 2015 году.

Учредитель и издатель

Федеральное государственное бюджетное образовательное учреждение высшего образования «Петербургский государственный университет путей сообщения Императора Александра I» (ФГБОУ ВО ПГУПС)

Адрес редакции:
190031, Санкт-Петербург, Московский пр., 9, ауд. 1–210
e-mail: itt-pgups@yandex.ru, сайт: http://itt-pgups.ru

ISSN 2413-2527

Журнал зарегистрирован Федеральной службой по надзору в сфере связи и массовых коммуникаций, свидетельство ЭЛ № ФС77-61707 от 07 мая 2015 г.

Журнал зарегистрирован в Российском индексе научного цитирования (РИНЦ).

Intellectual Technologies on Transport
Issue № 2, 2019

Network electronic scientific journal, open access. It publishes articles in Russian and English with the results of research and practical achievements in the field of intelligent technologies and associated research

Founded in 2015

Founder and Publisher

Federal State Educational Institution of Higher Education
«Emperor Alexander I Petersburg State Transport University»

Editorial adress:
190031, St. Petersburg, Moskovskiy pr., 9, aud. 1–210
e-mail: itt-pgups@yandex.ru, http://itt-pgups.ru

ISSN 2413-2527

The journal is registered by the Federal Service for Supervision of Communications and Mass Media, EL no. FS77-61707 testimony from May 7, 2015.

The journal is registered in the Russian Science Citation Index (RSCI).

Содержание

Contents

Справедливые дележи общего дохода в кооперативных играх

д.ф.-м.н. М.М. Луценко, к.т.н. А.М. Дёмин
Петербургский государственный университет путей сообщений Императора Александра I
Санкт-Петербург, Россия
ml4116@mail.ru

Аннотация. **Рассмотрены принципы построения дележей (платежей) участниками одного проекта, если разные участники имеют разный экономический и административный вклад в общий проект. Строятся кооперативные игры, учитывающие разный статус игроков. Обсуждаются ядро и вектор Шепли как примеры «справедливых» дележей. Для игр с частично упорядоченным множеством игроков, в которых функция выигрыша согласована с этим порядком, приведены аналитические формулы расчета вектора Шепли. Для приведенных примеров конфликтных ситуаций обсуждается «справедливость» построенных дележей.**

Ключевые слова: **кооперативная игра, вектор Шепли, частичный порядок, справедливое распределение дохода, распределение долей платежей.**

Введение

Всякое успешное предприятие (проект) начинается с продуманной кооперации участников. При совместной прокладке (эксплуатации) компьютерных сетей возникают проблемы, связанные с распределением общего дохода (платежа) между участниками проекта. Кооперация устойчива тогда, когда все участники согласны с принципами распределения их общего дохода (платежа) и понимают, какие доходы (платежи) они получат (выполнят) в результате заключенного соглашения. Построением «справедливых» дележей занимается теория кооперативных игр [1–5].

В этих моделях каждая коалиция участников проекта может отказаться от него и переключиться на другой проект с доходом, заранее известным всем участникам общего проекта. Поэтому при обсуждении договора каждая коалиция участников может обоснованно угрожать выйти из общего соглашения, что существенно затрудняет нахождение общего компромисса. В теории кооперативных игр предлагается несколько приемлемых выходов из конфликтной ситуации. Один из них — совместное принятие правила распределения доходов, основанного на идеях «симметрии и аддитивности»: равные участники кооперации должны получать равные доли общего дохода, а при одновременном участии в нескольких проектах доходы участников складываются. Несмотря на универсальность этих идей, построенное на них правило распределения доходов может приводить к совершенно неожиданным результатам, особенно в тех случаях, когда участники имеют разный экономический или административный вес.

В работе будет рассмотрено два принципа построения «справедливых» дележей: с-ядро и вектор Шепли [1–5]. Несмотря на их популярность, выполнение конкретных расчетов оказывается технически трудным, так как мы должны учитывать значения функции, заданной на множестве всех подмножеств множества участников проекта. Сложность формул затрудняет внедрение принципов «справедливости» при построении конкретных дележей.

В последние годы были выделены некоторые классы кооперативных игр, в которых вектор Шепли имеет простую структуру [4, 6, 7]. Это кооперативные игры, построенные по иерархическим системам. В них доходы участников зависят не только от их возможностей, но и от управляющих воздействий «более значимых» игроков. Аналитический вид вектора Шепли способствует тому, что дележ, построенный для подобных моделей, будет принят всеми участниками проекта.

Успешное применение аналитических выражений при построении оптимальных решений в сложных иерархических системах на транспорте можно найти в работе [8].

Кооперативные игры и их решения

Для построения моделей справедливого распределения доходов сформулируем основные определения и необходимые утверждения кооперативной теории игр.

Определение 1. Кооперативная игра — упорядоченная пара $\Gamma = \langle N, v \rangle$, в которой $N = \{1, 2, \ldots, n\}$ — множество игроков, а v — характеристическая функция, заданная на множестве всех подмножеств множества игроков N. Характеристическая функция v каждой коалиции K (подмножества игроков, $K \subseteq N$) ставит в соответствие выигрыш, который игроки могли бы получить, создав эту коалицию.

Величину $v(K)$ часто интерпретируют как доход участников проекта, объединенных в коалицию K. В некоторых случаях значение характеристической функции $v(K)$ можно интерпретировать как платеж, который должна выполнить коалиция K, если она будет создана [1–5]. Подобная ситуация может возникнуть при совместном строительстве компьютерной сети.

Определим те условия, при которых все игроки объединятся в большую коалицию N. Такое объединение возможно, если игроки договорятся о том, как они поделят общий выигрыш (платеж) $v(N)$.

Под дележом кооперативной игры $\Gamma = \langle N, v \rangle$ мы понимаем упорядоченный набор чисел $x = (x_1, x_2, \ldots, x_n)$, в котором компонента x_i — доля выигрыша (платежа), которую получит игрок i в соответствии с дележом x. Мы считаем, что все дележи коллективно рациональны, то есть сумма всех компонент каждого дележа равна $v(N)$.

Определение 2. *Ядром* классической кооперативной игры $\Gamma = \langle N, v \rangle$ называется множество дележей $C(v)$, для которых выполнены неравенства

$$x(K) = \sum_{i \in K} x_i \geq v(K) \text{ для всех } K \subset N.$$

Возможна следующая интерпретация введенного выше определения. Пусть коалиция K имеет возможность самостоятельно заключить на стороне контракт на сумму $v(K)$ (это число может равняться нулю). Допустим, что в результате предварительных переговоров предложен следующий дележ: компания с номером i получает величину x_i, причем $\sum_{i=1}^{n} x_i = v(N)$, $x_i \geq 0$. Если сумма величин x_i с номерами $i \in K$ (общая сумма, предлагаемая коалиции K) меньше величины $v(K)$, то компании, входящие в коалицию, откажутся от предложенного дележа на том основании, что у них есть более выгодное предложение. Таким образом, проблема состоит в нахождении вектора $(x_1, x_2, \ldots, x_n)$, описывающего дележ суммы контракта $v(N)$, удовлетворяющего все возможные коалиции, а ядро $C(v)$ — возможно пустое подмножество дележей, стабилизируемых простыми угрозами. Заметим, что все дележи в ядре индивидуально рациональны: $x_i \geq v(i)$ для всех $i \in N$.

Система неравенств, описывающая ядро, вместе с условием «коллективной рациональности» содержит $2^n - 1$ ограничений (столько же, сколько имеется в игре $\Gamma = \langle N, v \rangle$ непустых коалиций) и задает в n-мерном пространстве выпуклое замкнутое множество. Таким образом, множество дележей $C(v)$ определяется своими крайними точками в пространстве R^n.

В тех случаях, когда значение $v(K)$ характеристической функции интерпретируется как платеж коалиции K, определяют двойственное ядро $C^*(v)$ через систему противоположных неравенств [9]:

$$x(K) = \sum_{i \in K} x_i \leq v(K) \text{ для всех } K \subset N.$$

Двойственное ядро состоит из платежей, стабилизируемых простыми угрозами выхода из большой коалиции.

К недостаткам ядра следует отнести его возможное отсутствие, или оно может оказаться слишком обширным, тогда выбор реализуемого дележа из такого ядра может оказаться затруднительным.

Справедливые дележи

Множественность рассмотренного выше принципа решения кооперативной игры, а также жесткие условия существования ее ядра стимулируют попытки построения других принципов, применимых к каждой кооперативной игре. Одним из таких принципов является нормативный принцип распределения общего дохода (платежа) между участниками кооперативной игры. Для его применения необходимо, чтобы все игроки согласились на его применение до начала игры. Последовательное проведение принципов: равным игрокам равный выигрыш, выигрыши участников нескольких игр складываются, получает больше тот, кто больше вносит в коалицию, – приводит к дележу, «справедливому» по Шепли, или вектору Шепли.

Итак, под решением кооперативной игры мы понимаем функцию, ставящую в соответствие каждой кооперативной игре $\Gamma = \langle N, v \rangle$ из некоторого класса игр G вектор $Sh(v) = (Sh_1, Sh_2, \ldots, Sh_n)$, обладающий «разумными», с точки зрения всех игроков, свойствами. Компоненты вектора Шепли мы будем интерпретировать как полезности, получаемые игроками в результате соглашения или решения арбитра. Мы считаем, что наши соображения о справедливом дележе воплощены в следующих четырех аксиомах, впервые в несколько иной форме сформулированными Шепли в 1953 г. [10, 11].

Определение 3. Вектором Шепли называется отображение Sh, которое каждой кооперативной игре $\Gamma = \langle N, v \rangle$ с n игроков ставит в соответствие вектор $Sh(v) = (Sh_1, Sh_2, \ldots, Sh_n)$, обладающий следующими свойствами.

1. *Аксиома болвана.* Если игрок i таков, что для любой коалиции $K \subseteq N$ выполняется равенство $v(K \cup \{i\}) = v(K)$, то соответствующая компонента вектора Шепли равна нулю, то есть $Sh_i(v) = 0$.

Итак, если игрок i ничего не вносит ни в какую коалицию, то его доля от общего выигрыша равна нулю.

2. *Аксиома эффективности*. Сумма компонент вектора Шепли равна общему выигрышу всех игроков:

$$\sum_{i \in N} Sh_i(v) = v(N).$$

Таким образом, игроки делят между собой их общий доход (платеж).

3. *Аксиома симметрии.* Если игроки входят в игру одинаково, то соответствующие компоненты вектора Шепли равны.

4. *Аксиома агрегации (аддитивности).* Если характеристическая функция игры $\Gamma = \langle N, v \rangle$ равна сумме характеристических функций игр $\langle N, v_1 \rangle$ и $\langle N, v_2 \rangle$, то есть $v = v_1 + v_2$, то вектор суммы равен сумме векторов: $Sh(v_1 + v_2) = Sh(v_1) + Sh(v_2)$.

Эта аксиома утверждает, что выигрыш игрока, участвующего в двух играх одновременно, складывается из выигрышей в этих играх.

Теорема 1 (теорема Шепли о существовании справедливого дележа [10–11]). Существует, и притом только одно, отображение множества характеристических функций игр n лиц во множестве дележей, обладающее четырьмя перечисленными выше аксиомами, причем компоненты этого отображения можно найти по формуле:

$$Sh_i(v) = \sum_{K \subseteq I} \frac{(n-k-1)!k!}{n!}\big(v(K \cup \{i\}) - v(K)\big),$$

$$k = |K|;\ n = |N|,\ i \in N.$$

Вектор Шепли обладает целым рядом дополнительных свойств, подтверждающих его исключительность. Одно из них связано с математическим ожиданием дохода игрока.

Определение 4. Для игрока $i \in N$ в игре $\Gamma = \langle N, v \rangle$ величина $\Delta_i(K) = v(K \cup \{i\}) - v(K)$ называется *вкладом игрока i в коалицию K* или приращением выигрыша коалиции K за счет игрока i.

Величина $\Delta_i(K) = v(K \cup \{i\}) - v(K)$ – разумной платой за присоединение игрока i к коалиции K, если последняя уже сформирована.

Теорема 2 (вероятностное определение вектора Шепли [1–5]). Если игроки кооперативной игры $\Gamma = \langle N, v \rangle$ присоединяются к ней в случайном порядке, причем при своем присоединении i-й игрок получает вклад $\Delta_i(K)$, где K — коалиция, возникшая до его присоединения, то математическое ожидание выигрыша i-го игрока равно соответствующей компоненте вектора Шепли.

Все перечисленные свойства вектора Шепли говорят о том, что игроки посчитают справедливым распределение общего выигрыша (платежа) в соответствии с его компонентами.

Вектор Шепли для игр с иерархической структурой

Рассмотрим две модели, в которых игроки должны разделить общий доход (платеж) с учетом организационной иерархии и «справедливый» для всех ее сотрудников.

Пусть $N = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ – множество всех сотрудников организации. Для сотрудника α обозначим через $S(\alpha)$ множество его прямых подчиненных. Будем считать, что функция $S(\alpha)$ порождает на множестве сотрудников N иррефлексивный частичный порядок. Мы предполагаем, что сотрудник α приносит доход a_α в том и только в том случае, когда он получает разрешение от всех своих руководителей. Обозначим через $a = \{a_1, a_2, \ldots, a_n\}$ неотрицательный вектор доходов организации, причем через a_i обозначим доход, который может принести сотрудник α_i, $i = \overline{1,n}$.

Построим кооперативную игру $\Gamma = \langle N, v \rangle$, в которой игроками будут сотрудники рассматриваемой организации. По структуре организации и способу получения дохода построим характеристическую функцию игры $\Gamma = \langle N, v \rangle$. В частности, мы знаем, что коалиция $K \subseteq N$ получит доход a_α тогда и только тогда, когда эта коалиция содержит верхний замкнутый конус $\overline{V}[\alpha]$ (руководителей и самого участника). Следовательно, значение характеристической функции v коалиции K можно найти как сумму доходов всех участников коалиции. В результате мы получим:

$$v(K) = \sum_{\alpha \in N} u_{\overline{V}[\alpha]}(K),$$

где u_T — характеристическая функция простейшей игры с носителем T. В работе [6] приведены простые формулы для расчета вектора Шепли.

Пример 1. Предположим, что в предприятие входят четыре подразделения, имеющих разную степень самостоятельности. Каждое подразделение вносит свой вклад в общий доход предприятия, однако для правильной работы подчиненных подразделений необходимо разрешение всех их руководителей (см. схему). Обозначим через a_i тот вклад, который вносит подразделение i после получения разрешения руководителей. В частности, игроки 2 и 4 вносят (зарабатывают) a_2 и a_4 независимо от оставшихся подразделений. Подразделение 3 может привнести величину a_3, лишь получив разрешение (вступив в коалицию) от своего руководителя, подразделения 4. Подразделение 1 может привнести величину a_1, лишь получив разрешение (вступив в коалицию) от всех своих руководителей, подразделений 2, 3, 4. Предположим, что все подразделения согласились работать вместе. Рассчитаем дележ, справедливый по Шепли.

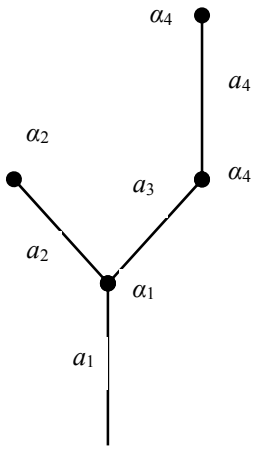


Схема иерархической структуры и выигрыши игроков

Запишем значения характеристической функции игры:

$$v_0 = 0,\ v_1 = v_3 = v_{1,3} = 0,\ v_2 = v_{1,2} = v_{2,3} = v_{1,2,3} = a_2,$$
$$v_4 = v_{1,4} = a_4,\ v_{3,4} = v_{1,3,4} = a_3 + a_4,\ v_{2,4} = v_{1,2,4} = a_2 + a_4,$$
$$v_{2,3,4} = a_2 + a_3 + a_4,\ v_{1,2,3,4} = a_1 + a_2 + a_3 + a_4.$$

Таким образом, игроки должны разделить заработанную ими вместе сумму: $v_{1,2,3,4} = a_1 + a_2 + a_3 + a_4$. Используя результаты работы [6], рассчитаем вектор Шепли. Его компоненты будут

$$Sh_1 = a_1/4\,,\ Sh_2 = a_2 + a_1/4\,,\ Sh_3 = a_3/2 + a_1/4\,,$$
$$Sh_4 = a_4 + a_3/2 + a_1/4\,.$$

Таким образом, подразделение 1 должно получать лишь четверть заработанной им суммы. Подразделение 2 в соответствии с согласованным принципом справедливости получит всю заработанную им сумму и четверть суммы, заработанной подразделением 1. Подразделение 3 получит половину своей заработанной суммы и четверть суммы, заработанной подразделением 1. Таким образом, административная надстройка оправдывает эксплуатацию подразделения 1.

Рассмотрим другой пример игры, в которой частичный порядок на множестве игроков согласован с их выигрышами. Пусть $N = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ — множество узлов компьютерной сети, частично упорядоченных иррефлексивным отношением. Множество узлов $S(\alpha)$, непосредственно предшествующих α, связано с узлом α линией стоимостью a_α, и узел α будет работать, если работают все узлы, предшествующие узлу α. Для работы сети с узлами из множества $K \subseteq N$ необходимо, чтобы работали все узлы, предшествующие узлам из K, то есть чтобы были построены соответствующие линии связи. Следовательно, стоимость строительства сети с узлами из K есть значение характеристической функции

$$v(K) = \sum_{\alpha \in V[K]} a_\alpha\,,$$

где $V[K]$ — множество узлов, предшествующих узлу из K, включая их самих. Простые формулы расчета компонент вектора Шепли были получены в работе [6].

Пример 2. Группа предпринимателей из четырех человек собирается построить водопроводную (электрическую) сеть по приведенной выше схеме. Обозначим через a_i стоимость части участка сети, обеспечивающей водой (электричеством) *i*-го предпринимателя. Таким образом, предпринимателю 1 достаточно построить участок стоимостью a_1. Предприниматель 2 получит воду лишь тогда, когда он построит участок стоимостью a_2 и подключится к уже построенному участку предпринимателя 1. Предприниматель 3 получит воду, если он построит участок стоимостью a_2 и подключится к водопроводу, построенному предпринимателем 1, а предприниматель 4 получит воду, построив участок стоимостью a_4 и подключившись к предпринимателю 3. Если предприниматели согласятся построить общую сеть в соответствии с предлагаемой схемой, то как они должны разделить платеж за строительство, используя вектор Шепли?

Запишем значения характеристической функции игры:

$$v_0 = 0\,,\ v_1 = a_1\,,\ v_2 = v_{1,2} = a_1 + a_2\,,\ v_3 = v_{1,3} = a_1 + a_3\,,$$
$$v_{2,3} = v_{1,2,3} = a_1 + a_2 + a_3\,,$$
$$v_4 = v_{1,4} = v_{3,4} = v_{1,3,4} = a_1 + a_3 + a_4\,,$$
$$v_{2,4} = v_{1,2,4} = v_{2,3,4} = v_{1,2,3,4} = a_1 + a_2 + a_3 + a_4\,.$$

Хотя эти игры имеют различные характеристические функции, однако они имеют одинаковые векторы Шепли [4]. Следовательно, игрок 1 должен заплатить лишь четверть стоимости первого участка, то есть $a_1/4$. Игрок 2 должен оплатить свой участок и четверть стоимости начального участка. Игрок 3 оплачивает половину стоимости своего участка и четверть стоимости первого участка. Игрок 4 оплачивает свой участок, половину участка игрока 3 и четверть стоимости игрока 1.

Заключение

Рассмотренные примеры позволяют оценить влияние организационных факторов на распределение доходов.

Литература

1. Воробьев Н.Н. Теория игр для экономистов-кибернетиков. — М. : Наука, 1985. — 272 с.
2. Теория игр : учебник. / Л.А. Петросян, Н.А. Зенкевич, Е. В. Шевкопляс. — 2-е изд., перераб. и доп. — СПб. : БХВ-Петербург, 2012. — 432 с.
3. Луценко М.М. Теория статистических решений : учеб. пособие. Ч. 2. — СПб. : ПГУПС, 2012. — 110 с.
4. Луценко М.М. Теория игр : учебное пособие / М.М. Луценко, А.М. Дёмин. — СПб. : ПГУПС, 2018. — 71 с.
5. Мазалов В. В. Математическая теория игр и приложения : учеб. пособие / В. В. Мазалов. – СПб. : Лань, 2010. — 446 с.
6. Луценко М.М. Веса Шепли для заданий педагогического теста / М.М. Луценко, Н.В. Шадринцева // Вестник Санкт-Петербургского университета. Прикладная математика. Информатика. Процессы управления. – 2017. – Т. 13. – № 3. – С. 300–312.
7. Луценко М.М. Принятие инвестиционных решений в строительстве при неполной информации о функционировании объекта / М.М. Луценко, А.М. Дёмин // Управление рисками в экономике: проблемы и решения / отв. ред. С.Г. Опарин. — СПб. : Санкт-Петербургский политехнический университет Петра Великого, 2015. – С. 237–259.
8. Модели управления рисками и ресурсами автоматизированных систем критического применения железнодорожного транспорта с учетом экономического фактора / А.А. Корниенко, С.Е. Ададуров, А.П. Глухов, С.В. Диасамидзе, В.Н. Кустов // Известия Петербургского университета путей сообщения. – 2017. – Т. 14. – № 4. – С. 588–596.
9. Maschler M., Solan E., Zamir S. Game Theory. Translated from the Hebrew by Ziv Hellman; English ed. by Mike Borns. Cambridge, Cambridge University Press, 2013, 1009 p.
10. Печерский С.Л., Яновская Е.Б. Кооперативные игры: решения и аксиомы. – М. : Европейский Университет в Санкт-Петербурге, 2004. — 459 с.
11. Shapley L.S. A value of n-person games. *In: H.W. Kuhn, A.W. Tucker, eds., Contributions to the Theory of Games*, 1953, Vol. 2, Princeton, Princeton University Press, pp. 307–317.

Fair Sharing`s of Total Income at Cooperative Games

Grand PhD M.M. Lutsenko, PhD A.M. Demin
Emperor Alexander I Petersburg State Transport University
St. Petersburg, Russia
ml4116@mail.ru

Abstract. **The paper considers the principles of construction of distribution (imputations) by the participants of the same project, if different participants have different economic and administrative contributions to the overall project. Cooperative games are built, taking into account the different status of the players. The core and the Shapley vector are discussed as examples of «fair» distributions. For games with a partially ordered set of players, analytical formulas for calculating the Shapley vector are given. For the considered examples of conflict situations, the «fair» of the constructed distributions is discussed.**

Keywords: **TU-game, Shapley value, partial order, fair distribution of income, distribution of shares of payments.**

REFERENCES

1. Vorob'yev N.N. Game theory for cybernetic economists [Teoriya igr dlya ekonomistov-kibernetikov], Moscow, The Science, 1985, 272 p.

2. Petrosyan L.A., Zenkevich N.A., Shevkoplyas E.V. Game theory: Textbook [Teoriya igr: Uchebnik], St. Petersburg, BHV-Peterburg, 2012, 432 p.

3. Lutsenko M.M. Statistical decision theory: Study guide. Part 2 [Teoriya statisticheskikh resheniy: Uchebnoe posobie. Chast` 2], St. Petersburg, PSTU, 2012, 110 p.

4. Lutsenko M.M., Demin A.M. Game theory: Study guide [Teoriya igr: Uchebnoe posobie], St. Petersburg, PSTU, 2018, 71 p.

5. Mazalov V.V. Mathematical game theory and its application: Study guide [Matematicheskaya teoriya igr i prilozheniya: Uchebnoe posobie], St. Petersburg, LAN, 2010, 446 p.

6. Lutsenko M.M., Shadrinceva N.V. Shapley Weights for Test Items. [Vesa Shepli dlya zadaniy pedagogicheskogo testa], *Vestnik of Saint Petersburg University. Applied mathematics. Computer Science. Control Processes [Vestnik Sankt-Peterburgskogo universiteta. Prikladnaya matematika. Informatika. Protsessy upravleniya]*, 2017, Vol. 13, No. 3, pp. 300–312.

7. Lutsenko M.M., Demin A.M. Investment Decisions — Making in Construction with Incomplete Information About the Facility's Operation [Prinyatie investitsionnykh resheniy v stroitel`stve pri nepolnoy informatsii o funktsionirovanii ob`ekta], *Risk Management in the Economy: Problems and Solutions [Upravlenie riskami v ekonomike: problemy i resheniya]*, St. Petersburg, Publishing House of Polytechnical University, 2015. pp. 237–259.

8. Korniyenko A.A., Adadurov S.Y., Glukhov A.P., Diasamydze S.V., Kustov V.N. Models of Risk and Resource Management of Railway Transport Critical Application Automation Systems with Regard to Economic Aspect [Modeli upravleniya riskami i resursami avtomatizirovannykh sistem kriticheskogo primeneniya zheleznodorozhnogo transporta s uchetom ekonomicheskogo faktora], *Proc. of Petersburg Transport University [Izvestiya Peterburgskogo universiteta putey soobshcheniya]*, 2017, Vol. 14. No. 4, pp. 588–596.

9. Maschler M., Solan E., Zamir S. Game Theory. Translated from the Hebrew by Ziv Hellman; English ed. by Mike Borns. Cambridge, Cambridge University Press, 2013, 1009 p.

10. Pecherskiy S.L., Yanovskaya E.B. Cooperative games: solutions and axioms [Kooperativnye igry: resheniya i aksiomy], Moscow, Publishing House of the European University in St. Petersburg, 2004, 459 p.

11. Shapley L.S. A value of n-person games. *In: H.W. Kuhn, A.W. Tucker, eds., Contributions to the Theory of Games*, 1953, Vol. 2, Princeton, Princeton University Press, pp. 307–317.

Унифицированная модель многоуровневых угроз нарушения информационной безопасности в сетях с динамической топологией

Р.А. Демидов, д.т.н. П.Д. Зегжда
Санкт-Петербургский политехнический университет Петра Великого
Санкт-Петербург, Россия
rd@ibks.spbstu.ru

***Аннотация*. Рассматривается задача выявления угроз кибербезопасности в транспортных сетях с динамической топологией (VANET, FANET, MARINET, MANET и др.). Показано наличие проявлений угроз на программном и сетевом уровнях представления системы, в связи с чем представлена модель угроз в этих сетях. Предложен нейросетевой подход по унифицированному выявлению таких угроз, экспериментально подтверждена способность подхода к выявлению обоих типов угроз.**

***Ключевые слова:* анализ киберугроз, глубокое обучение, гибридная нейронная сеть, динамическая сеть, VANET, FANET, MARINET, MANET.**

Введение

В настоящее время наблюдается повсеместное проникновение современных сетевых технологий в различные сферы человеческой жизнедеятельности. Одной из них является новая сфера транспортных сетей, среди которых выделяются сети автотранспорта [1] (автонет, VANET), летательных и плавательных средств (аэронет, FANET, маринет, MARINET соответственно), а также прочих модификаций. Основой таких сетей является подвижная беспроводная одноранговая компьютерная сеть с динамической топологией (СДТ), которая обладает ключевыми особенностями, такими как:

- *динамический характер топологии*: структура сети изменяема с течением времени по мере добавления и удаления сетевых устройств, а также в силу их перемещений;
- самоорганизация *и адаптивность*: архитектура СДТ работает автономно и динамически реагирует на изменения связности узлов. Алгоритмы маршрутизации в СДТ позволяют перестраивать маршруты в сети при быстром изменении её топологии;
- децентрализация: СДТ рассчитана на работу в условиях отсутствия выделенных узлов-маршрутизаторов;
- *равнозначность узлов*: каждый сетевой узел исполняет роли как отправителя и получателя информации, так и маршрутизатора. Маршруты между узлами пролегают через несколько промежуточных узлов.

Перечисленные свойства в совокупности обеспечивают устойчивость СДТ к изменениям и перемещениям сетевых узлов. Динамический характер топологии рассматриваемых сетей не позволяет использовать традиционные подходы к маршрутизации данных. Это, а также двойственность сетевой роли вершин СДТ требует использования специальных протоколов маршрутизации. На сетевом уровне в СДТ применяются протоколы динамической маршрутизации, которые условно можно разделить на два класса:

- проактивные (DSDV, OLSR): маршруты строятся для всей сети и распространяются каждому узлу;
- реактивные (AODV, DSR): маршруты строятся по требованию.

На сегодняшний день наиболее распространенным на практике алгоритмом маршрутизации является реактивный протокол AODV [2]. AODV остается эффективным в случае СДТ больших размеров, длинных путей и частых перестроений маршрутов. Кратчайшие пути от источника к цели строятся «по требованию», что позволяет поддерживать только востребованные маршруты и избежать излишней перегруженности сети.

Многоуровневый характер угроз в сетях с динамической топологией

Характерные особенности СДТ формируют новые условия для реализации в СДТ различных угроз нарушения безопасности (УНБ). Общими причинами их возникновения являются:

- общая физическая среда передачи данных: позволяет пассивное прослушивание эфира злоумышленником;
- общедоступность устройств: отсутствие встроенных средств защиты узлов от целенаправленных воздействий на сетевой стек устройств со стороны злоумышленников;
- отсутствие стандартизации, централизованного управления и системы доверенной верификации узлов, что позволяет нахождение в сети как изначально вредоносных устройств, так и легитимных устройств с уязвимым управляющим ПО;
- особенности реактивной маршрутизации (доверие к новым узлам «по умолчанию», отсутствие встроенных процедур установления доверия между узлами, двойственность сетевой роли узлов);

- невозможность реализации единой политики безопасности ввиду особенностей классической архитектуры СДТ, таких как отсутствие фиксированной топологии и центральных узлов.

Потенциальный злоумышленник может реализовать угрозу нарушения безопасности и целенаправленно воздействовать на СДТ с целью совершения вредоносных действий. Упомянутые УНБ имеют различную природу (рис. 1) и проявляются сразу на нескольких уровнях представления сети.

На программном уровне. Используя уязвимости управляющего программного обеспечения (ПО) у отдельных устройств СДТ, злоумышленник может реализовать различные вредоносные сценарии: несанкционированный доступ и сбор информации, фальсификация и уничтожение данных, вывод устройства из строя, заражение вредоносным кодом с целью дальнейшего развития атаки на другие устройства СДТ.

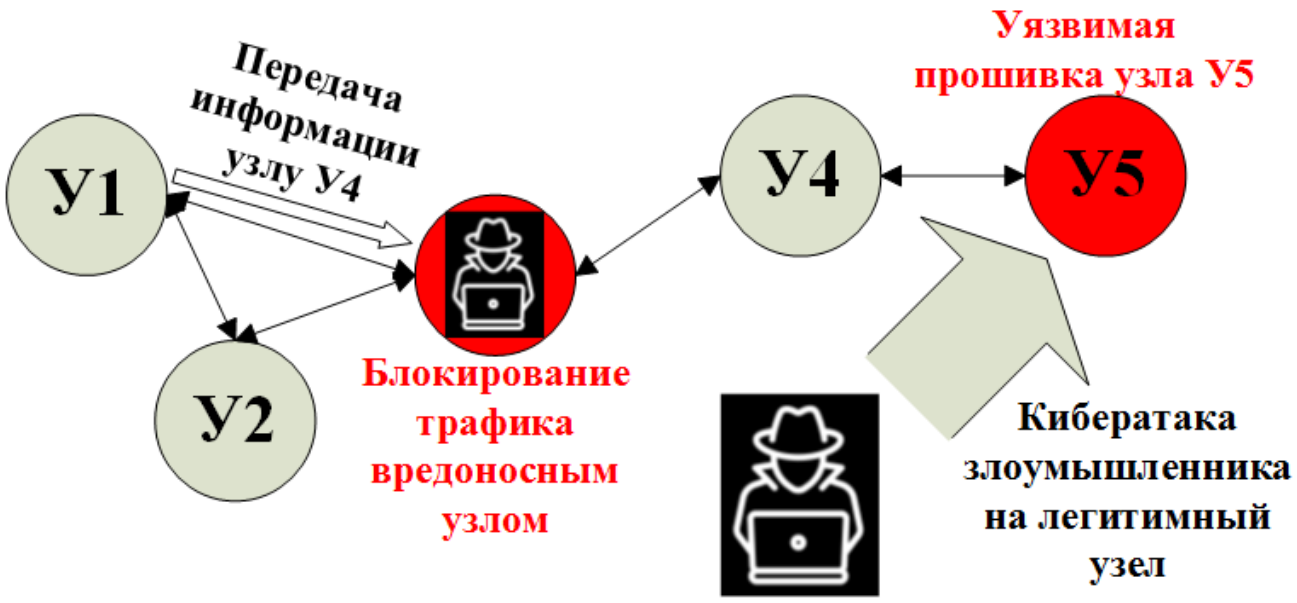


Рис. 1. Многоуровневый характер проявления УНБ в СДТ

На программном уровне представления система может быть описана графами передачи управления целевой программы с базовыми блоками кода в качестве вершин и с инструкциями передачи управления в качестве ориентированных ребер. В случае самомодифицирующегося кода программа представляется несколькими такими графами. Среди УНБ программного уровня можно выделить программные уязвимости типа «переполнение буфера», «двойное освобождение памяти» и др.

На сетевом уровне. Такие УНБ связаны с дезорганизацией связности сети и вызваны воздействием на проходящий трафик со стороны отдельных изначально вредоносных узлов при сетевом обмене узлов друг с другом. Данные угрозы проявляются в нарушении маршрутизации и топологии СДТ, в выводе из строя узлов, что приводит к замедлению, блокированию и перехвату трафика, хищениям и искажениям информации.

На сетевом уровне система представляется в виде изменяющегося во времени графа связности различных устройств СДТ. Среди УНБ сетевого уровня выделяются угрозы типов «черная дыра» [3], «червоточина» [5] и др.

Таким образом, в обоих случаях может быть сформировано обобщенное описание системы в форме изменяющегося во времени графа. Основываясь на этом, в работе предлагается унифицированный подход к выявлению как сетевых, так и программных угроз нарушения безопасности в СДТ.

Модель многоуровневых угроз и нейросетевой метод их выявления в сетях с динамической топологией

В рамках подхода предлагается унифицированное определение наличия УНБ в СДТ, представленной графом передачи управления программы либо изменяющимся графом связности СДТ. Наличие в системе угрозы обусловлено наличием в ней некоторой комбинации скрытых факторов (свойств, признаков), которые являются статистическими и подчас не имеют строгой формализации (рис. 2) (1).

$$\forall x\colon \exists\text{УНБ}(x) == h(\{\text{скрытые факторы системы } x\}_i)$$

$$\forall x\colon \text{скрытый фактор } i \text{ системы } x = \\ = g_i(\text{описание системы } x) \qquad (1)$$

Таким образом, наличие в системе многоуровневых угроз определяется значением введенной *булевой функции существования угрозы* T (2), (3).

$$T(x) = h(\{\text{скрытые факторы системы } x\}_i) = \\ = h(\cup_i g_i(\text{описание системы } x)) \qquad (2)$$

$$S = S_{program} \cup S_{graph} \quad T\colon S \to \{0,1\}. \qquad (3)$$

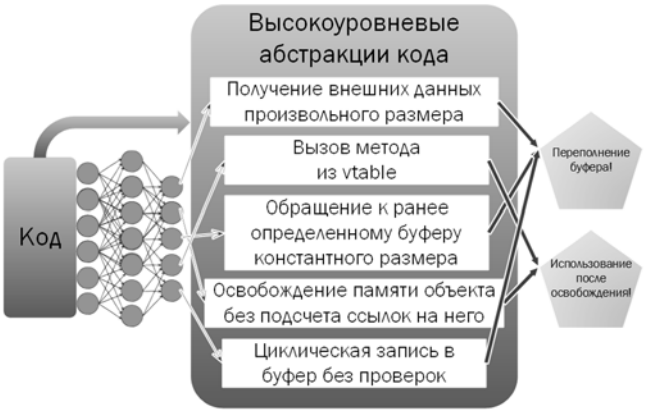


Рис. 2. Программные уязвимости как комбинация высокоуровневых свойств кода

Аналитическое построение искомой функции T затруднено в силу невозможности формализации признаков наличия угроз системе, в силу недостатка информации о неизвестных угрозах, а также в силу ограниченного характера знаний об известных угрозах.

Предлагается подход, заключающийся в построении приближения к булевой функции существования угрозы T с использованием глубокой нейронной сети специальной конфигурации. Такая сеть обучается на основе известных данных об угрозах и обеспечивает работу с входными данными в обобщенном виде как временного ряда ориентированных графов, что позволяет описать систему как на программном, так и на сетевом уровне представления.

Предлагаемая структура нейросети представлена на рисунке 3. Ее структура содержит графовую сверточную и рекуррентную части.

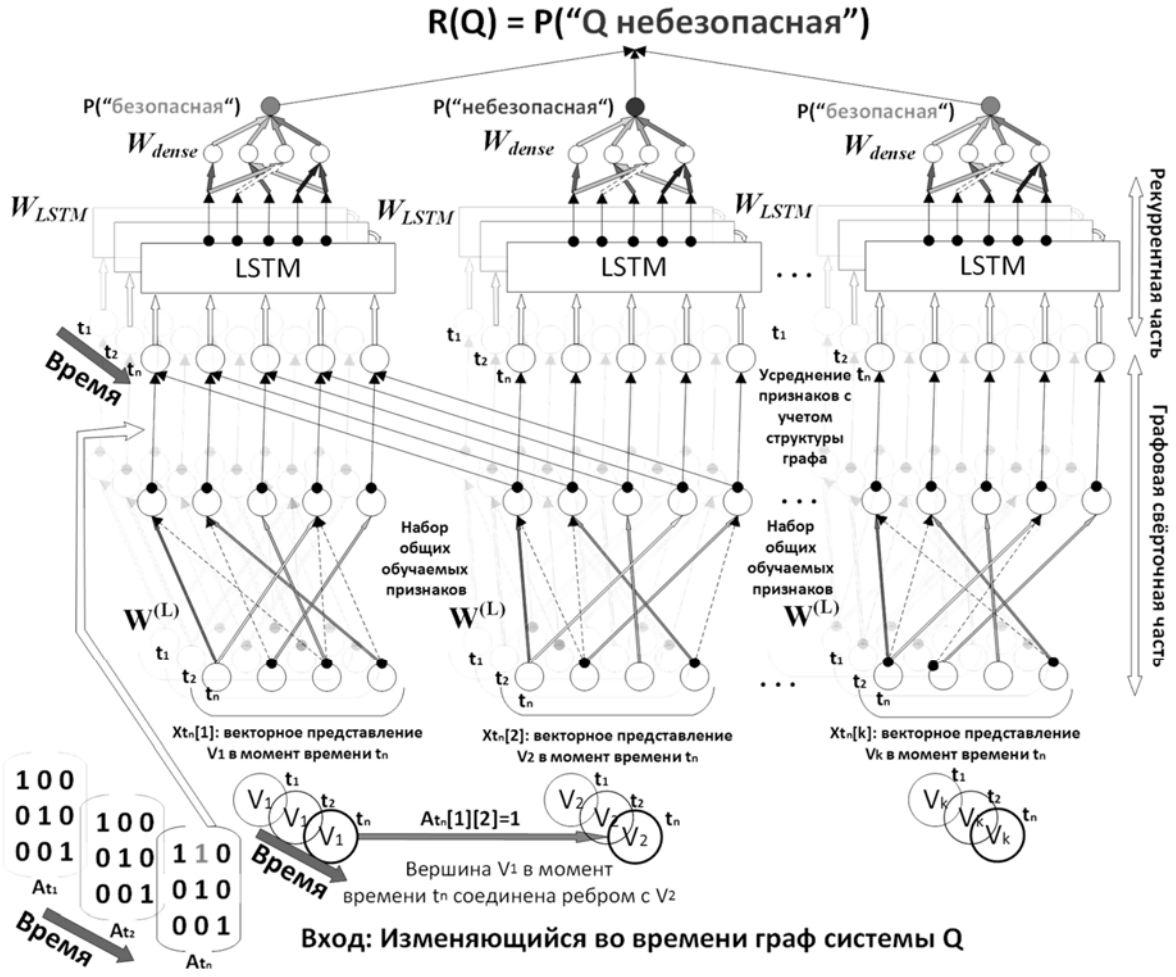


Рис. 3. Структура слоев гибридной ГНС для унифицированного выявления угроз в СДТ

При обучении нейросети на тренировочном наборе известных данных об угрозах СДТ происходит подстройка данных коэффициентов, с тем чтобы наилучшим образом предсказывать наличие угрозы на примерах из обучающего множества. После фазы обучения данные коэффициенты фиксируются. Обученная нейросеть при поступлении на вход описания новой системы вычисляет приближение к булевой функции существования угрозы, что позволяет оценить наличие угрозы для новых систем, не входящих в обучающую выборку.

Практические результаты подхода

Предлагаемый в работе подход был реализован на практике для различных типов сетевых и программных угроз.

Для программных угроз, которые позволяют реализовать угрозу «отказ в обслуживании» узлов СДТ через эксплуатацию уязвимости разыменования нулевого указателя в машинном коде, реализация подхода позволила достигнуть точности в 98 % в режиме распознавания. Для программных угроз, которые позволяют реализовать угрозы перехвата управления, заражения вредоносным кодом, повышения привилегий на узлах СДТ через эксплуатацию уязвимостей двойного освобождения, использования после освобождения и т. д., технология позволяет повысить точность до значений 92…99 % для разных типов программных угроз (рис. 4, *а*).

а) б)

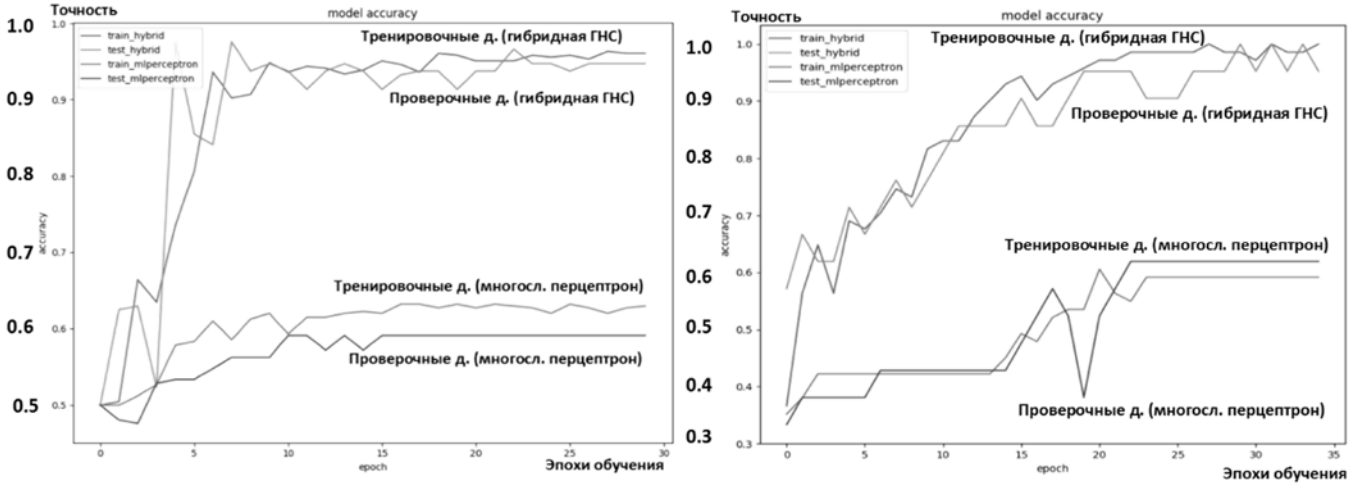


Рис. 4. Прирост точности гибридной сети по сравнению с классической нейросетевой архитектурой: *а* — для угроз на программном уровне; *б* — для угроз на сетевом уровне

Для сетевых угроз типов «черная дыра», «червоточина» и т. д., которые позволяют реализовать блокирование, фальсификацию, искажение сетевого трафика, разработка позволяет достигнуть точности выявления угроз в режиме распознавания от 95 до 98 % (рис. 4, *б*).

Выводы

В работе представлен подход к моделированию угроз в сетях с динамической топологией в условиях многоуровневого характера их проявления и недостатка известных данных. Использование глубоких нейронных сетей позволяет на практике осуществлять унифицированное выявление программных и сетевых угроз в СДТ с высокой достоверностью.

Литература

1. Shankar, R., Singh, A.V. Use of VANETs for Human Safety in Road Transportation, *4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions)*, Noida, 2015, pp. 1–6.

2. Карманов М. Л. Протокол маршрутизации для ad-hoc сетей // Вестник Южно-Уральского государственного университета. Сер. Компьютерные технологии, управление и радиоэлектроника. – 2009. – № 26 (159). – С. 47–51.

3. Christey, S., Martin, R. A. Vulnerability Type Distributions in CVE. Mitre Report, May 2007. Available at: http://cwe.mitre.org/documents/vuln-trends.html.

4. Tseng, F.-H., Chou, L.-D. and Chao H.-C. A Survey of Black Hole Attacks in Wireless Mobile Ad Hoc Networks, *Journal on Human-Centric Computing and Information Sciences Springer*, 2011, Vol. 1, No. 4, pp. 1–16.

5. Maulik, R., Chaki, N. A Study on Wormhole Attacks in MANET. *Proceeding of International Journal of Computer Information Systems and Industrial Management Applications*, 2011, Vol. 3, pp. 271–279.

Unified Model of Multilevel Security Threats in Networks with Dynamic Topology

R.A. Demidov, Grand PhD P.D. Zegzhda
Peter the Great St. Petersburg Polytechnic University
Saint Petersburg, Russia
rd@ibks.spbstu.ru

Abstract. The **article deals with the problem of identifying cybersecurity threats in transport networks with dynamic topology (VANET, FANET, MARINET, MANET, etc.). The presence of threats both on the network and software levels of the system is shown, and therefore the model of threats is presented. A neural network approach for unified detection of such threats is proposed and the ability to identify both types of threats is experimentally confirmed.**

Keywords: **cyber threat analysis, deep learning, hybrid neural network, dynamic network, VANET, FANET, MARINET, MANET.**

References

1. Shankar, R., Singh, V.A. Use of VANETs for Human Safety in Road Transportation, *4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions)*, Noida, 2015, pp. 1–6.

2. Karmanov M. L. Routing in Ad-Hoc Networks [Protokol marshrutizatsii dlya ad-hoc setey], *Bulletin of the South Ural State University. Series: Computer Technologies, Automatic Control & Radioelectronics [Vestnik Yuzhno-Ural'skogo gosudarstvennogo universiteta. Seriya: Komp'yuternye tekhnologii, upravlenie i radioelektronika]*, 2009, No. 26 (159), pp. 47–51.

3. Christey, S., Martin, R. A. Vulnerability Type Distributions in CVE. Mitre Report, May 2007. Available at: http://cwe.mitre.org/documents/vuln-trends.html.

4. Tseng, F.-H., Chou, L.-D. and Chao H.-C. A Survey of Black Hole Attacks in Wireless Mobile Ad Hoc Networks, *Journal on Human-Centric Computing and Information Sciences Springer*, 2011, Vol. 1, No. 4, pp. 1–16.

5. Maulik, R., Chaki, N. A Study on Wormhole Attacks in MANET. *Proceeding of International Journal of Computer Information Systems and Industrial Management Applications*, 2011, Vol. 3, pp. 271–279.

About One Approach to Creation of Information-Secure Communication Systems

PhD V.B. Vilkov
Military Academy of Logistics
of the Ministry of Defence of the
Russian Federation
St. Petersburg, Russia
amirusha@rambler.ru

Grand PhD A.K. Chernykh
St. Petersburg Military Institute
of the National Guard of the
Russian Federation
St. Petersburg, Russia
nataliachernykh@mail.ru

PhD A.I. Dergachev
Emperor Alexander I
St. Petersburg State
Transport University
St. Petersburg, Russia
d_ader@mail.ru

Summary. **The article deals with the approach to the creation of communication systems, to the maximum extent providing information security of the data processed within them. We propose an optimal algorithm that implements this approach, which is illustrated by an example. Given the problem statement, implementing the creation of these communications systems, based on the ideas of multi-objective optimization.**

Keywords**: information security of the system, reliability of the communication system, theory of fuzzy sets, fuzzy logic, edges of graphs, maximum skeleton.**

INTRODUCTION

The article deals with the task of creating the most effective, according to the criterion of information security (hereinafter reliable), communication system. This problem is a modification of the known problem of the shortest connection [1–4]. The difference is that as a characteristic of the communication channel is used not its cost, but its reliability (information security), which involves the theory of fuzzy sets and fuzzy logic [5–14].

It should be noted that with the help of this task Rossi, heiser and king proposed a scheme for laying television cables connecting all stations into a single network [15]. In addition, the proposed problem is relevant in a number of other cases when you want to link certain nodes (points) with the least effort and money, that is, to build a minimum network. In [16], an example is considered in which the probability of information loss during its use is used as a characteristic of the communication channel. It is noted in the literature that some problems lead to the need to build a network of maximum and not minimum weight. This task is also applicable algorithm Kruskal, if you change the sign of the weight of each edge to the opposite. If you want to find a network with a minimal product of edge weights, then given that log(ab) = log(a) + log(b), the minimal main tree of the graph in which the edge weights are replaced by their logarithms gives the desired solution. However, the weight of the ribs must be positive.

We give the formulation of the problem in the language of graph theory [2, 3]. Edge, which is mapped to a number (length, weight, capacity, reliability, etc.) will be measured. Graph, all edges of which are weighted, we call balanced. Let's call the skeleton of a connected graph *G* subgraph, which is a tree and contains all the vertices of the graph *G*. You want to find the most reliable skeleton.

The formulated task is particularly relevant when creating communication networks, when it is important to minimize hackers ' access to information circulating in them.

It is natural to understand the reliability of the communication channel as we are confident in the safety of its use in the course of information exchange. As an indicator of the reliability of the communication channel, we can use the probability that there will be no negative consequences due to the impact on it from hackers, natural and man-made disasters in the course of information exchange. But here a serious question arises-where to get these probabilities, especially if the parameters of the functioning of the communication channel are confidential, which is quite natural. In addition, even if these probabilities are known, it remains a rather difficult technical task to determine the optimal use of the entire communication network.

We propose to use the theory of fuzzy sets to determine the reliability of communication channels and the entire communication network. Recall the necessary for further concepts of this theory.

The concept of a fuzzy set is an attempt of mathematical formalization of fuzzy information for the construction of mathematical models. At the heart of this concept is the idea that the elements that make up a given set, having a common property, can have this property to a different extent and, therefore, belong to this set with different degrees. In this approach, saying that some element belongs to a given set, it is necessary to specify the degree to which this element satisfies the properties of this set.

A fuzzy set $\widehat{A}$ on a universal set U is a set of pairs $\left(\mu_{\widehat{A}}(u),u\right)$, where $\mu_{\widehat{A}}(u)$ — the degree of membership of the element $u \in U$ to the fuzzy set $\widehat{A}$. The function $\mu_{\widehat{A}}(u)$ is called the fuzzy membership function $\widehat{A}$, $\mu_{\widehat{A}}(u)$ expresses the degree of membership of the element $u \in U$ to the fuzzy set $\widehat{A}$. The degree of belonging is a number from the interval [0, b]. The higher the degree of membership, the more the element of the universal set corresponds to the properties of the fuzzy set, the more reliably we can say that it is an element of this set.

As a rule, it is assumed that the membership function takes values from the interval [0, 1]. Questions related to the

definition of the type of membership function and their construction are studied, for example, in [10, 17].

In the future, we will often use the words «degree of belonging», «information security», «reliability» instead of the phrase «meaning of the function of belonging».

Definitions of fuzzy set-theoretic operations of Union, intersection and others can be generalized from the usual set theory. We present the definitions of fuzzy set-theoretic intersection and Union operations proposed by L. Zadeh.

The intersection of fuzzy sets $\widehat{A}$ and $\widehat{B}$ given on U is called the fuzzy set $\widehat{C} = \widehat{A} \cap \widehat{B}$ with the membership function

$$\mu_{\widehat{C}}(u) = min\left\{\mu_{\widehat{A}}(u), \mu_{\widehat{B}}(u)\right\} \qquad (1)$$

for all $u \in U$ (see fig.1).

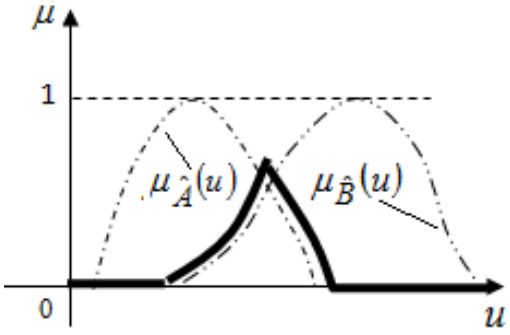


Fig. 1. Intersection of fuzzy sets

Fuzzy sets in the case where the universal set is a numerical axis, called fuzzy quantities. A fuzzy quantity whose membership function is continuous and has a single maximum is called a fuzzy number.

Following [18–20], we introduce some concepts of fuzzy logic. In classical mathematical logic, the values of the truth of statements can be only two values – «true» and «false», with the value of «true» corresponds to the number 1, the value of «false» – the number 0. Fuzzy logic deals with fuzzy statements that may be true or false to some extent. The degree of truth of a fuzzy statement takes values from a closed interval, with 0 coincides with the value of «false», 1 – with the value of «true».

The degree of truth of a fuzzy statement $\widetilde{F}$ is denoted by $\mu\left(\widetilde{F}\right)$.

Various logical operations are introduced on fuzzy statements, let us focus on two of them: conjunctions and disjunctions.

Consider two fuzzy statements $\widetilde{A}$ and $\widetilde{B}$. Fuzzy logical operations AND ($\wedge$) and OR ($\vee$) by analogy with the set-theoretic operations of Union and intersection performed by the rules:

$$\mu\left(\widetilde{A} \wedge \widetilde{B}\right) = \min\left\{\mu\left(\widetilde{A}\right), \mu\left(\widetilde{B}\right)\right\} \qquad (3)$$

The operation of finding the minimum is also indicated by a sign $\wedge$, i.e.

$$\mu_{\widehat{C}}(u) = \mu_{\widehat{A}}(u) \wedge \mu_{\widehat{B}}(u).$$

The Union of fuzzy sets $\widehat{A}$ and $\widehat{B}$ given on U is called a fuzzy set $\widehat{D} = \widehat{A} \cup \widehat{B}$ with the membership function

$$\mu_{\widehat{D}}(u) = \max\left\{\mu_{\widehat{A}}(u), \mu_{\widehat{B}}(u)\right\} \qquad (2)$$

for all $u \in U$ (see fig.2).

The operation of finding the maximum is also indicated by a sign $\vee$, i.e.

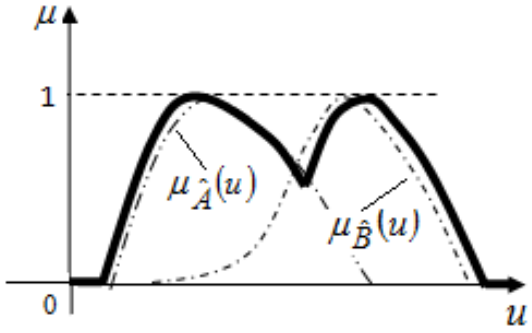


Fig. 2. The Union of fuzzy sets

$$\mu\left(\widetilde{A} \vee \widetilde{B}\right) = \max\left\{\mu\left(\widetilde{A}\right), \mu\left(\widetilde{B}\right)\right\} \qquad (4)$$

Under the reliability of the communication channel, we understand the degree of truth of the fuzzy statement «the communication channel is reliable». It is assumed that this indicator is known for any of the communication channels that can be created. By determining the conjunction of fuzzy statements (formula (3)), the reliability (degree of truth) of the network is equal to the minimum of the reliability (degrees of truth) of the communication channels included in this network.

So, let's give a connected weighted graph $G = (V, E)$, its edges correspond to the communication channels, each of them is correlated with a number – the degree of truth of the fuzzy statement «the communication channel is reliable».

It is required to find the skeleton of this graph, for which the degree of truth of the fuzzy statement «all communication channels are reliable» is maximal.

We will solve the problem using the modified Kruskal algorithm, replacing it only with the criterion of joining the next edge to the already constructed (perspective) set of edges, and we will look for not the minimum, but the maximum skeleton (the skeleton of the maximum weight). Denote the maximum skeleton will be MakO.

Theorems 1 and 2 take place for the case under consideration.

Theorem 1. Among the solutions of the problem of the most reliable subgraph containing all edges of the graph under consideration there is a skeleton.

Evidence. Suppose the opposite, i.e. that the subgraph which is the solution of the problem (connecting all vertices of the graph under consideration and having the maximum weight) is not a tree. Then there is a cycle in it. Removing an arbitrary link of this cycle, we get a subgraph containing all vertices, but not less weight. If the obtained subgraph is a tree, then we have obtained the required skeleton, otherwise there is a cycle in this subgraph, removing an arbitrary edge in it, obtaining a new subgraph of no less weight, etc. By virtue of finiteness of the number of edges in the original graph, we eventually obtain the required skeleton. The theorem is proved.

Changing with regard to the concept of «weight» algorithm Kruskal, we obtain the following algorithm.

Let a connected graph $G=(V,E)$ with *n* vertices and *m* edges be given, let its weight $e\in E$ be defined for any edge $d(e)$.

We start with a graph $G_0=(V,\varnothing)$ that consists only of the vertices of the graph $G=(V,E)$ and has no edges. This graph can be considered as *n* connected component, each of which consists of one vertex.

In the future, the algorithm consists of a sequence of stages. At the stage with the number $k=1,2,\ldots,n-1$ the graph is built $G_k=(V,T_k)$, for this purpose one edge is added to the set of edges T_{k-1} of the graph G_{k-1}, which is selected by the following rule:

– in the graph $G=(V,E)$, select the edge of the maximum weight of the number of edges that do not belong T_{k-1};

– if the addition of this edge to does T_{k-1} not lead to the formation of a cycle, then we attach this edge to T_{k-1}, we obtain T_k and graph $G_k=(V,T_k)$;

– if the cycle is formed, then of the remaining edges of the graph $G_k=(V,T_k)$ do not belong T_{k-1}, select the edge of the maximum weight, etc.

The graph G_{n-1} is the skeleton of a graph G of maximum weight.

Theorem 2. The stated algorithm gives MakO. (For a classic formulation see [4]).

Evidence. We show first that when $0<i<n-1$ a graph G_i can be constructed. Indeed, we consider the sets of edges T_{i-1} and T_{i-1}. Valid to the connectivity of the graph G, there is such an edge (k,l), as well $k\in T_{i-1}$ and $E\setminus T_{i-1}$. This edge does not form a cycle with edges from T_{i-1}. Having chosen an edge with the maximum weight from all such edges, we get an edge T_{i-1} whose joining gives T_i.

We prove now that T_{n-1} is the skeleton of the maximum weight in the graph G.

Consider a graph G_{n-1} with a set of edges T_{n-1}. Since it is connected, consists of *n* vertices, $n-1$ edges and has no cycles, it is a tree (see [21]). We show that the weight of the tree G_{n-1} is maximal.

Suppose that this is not so. Among all the skeletons of the graph G having the maximum weight, choose such a skeleton with a set of edges T that has T_{n-1} the maximum number of common edges.

Let $e_i=(a,b)$ be an edge T_{n-1} from that is not contained in T and has a minimum number among the edges of the set T_{n-1} that are not included in T. (It is assumed that the edges in the set T_{n-1} received numbers in the process of its construction, in the order of their joining to the graph under construction). In the set T there is a simple chain connecting vertices *a* and b by attaching an edge e_i to it we get a cycle. In this cycle, there is an edge *e* that is not included in T_{n-1}. Replacing an T edge *e* on e_i, get a new skeleton $T'=T\setminus\{e\}\cup\{e_i\}$. But T - the skeleton of a maximum weight, thus the weight T' not more weight T. It follows that the edge e_i does not weigh more than the edge *e*.

On the other hand, attaching an edge *e* to T_{i-1} (at $i=1$ assuming we $T_{i-1}=\varnothing$) do not get a cycle, since the edges $e_1,e_2,\ldots,e_{i-1},e$ are included in the set T. If the weight of the edge *e* was greater than the weight of the edge e_i, then when building a tree T_i, we would not e_i, and *e* (or another edge with a weight of more weight e_i). Therefore, the weight of an edge e_i is equal to the weight of an edge *e*, and the weights of trees with sets of edges T and T' are the same.

So, T' — the skeleton of the maximum weight. The number of edges common to sets T' and T is greater than the number of common edges for T_{n-1} and T ($T'=T\setminus\{e\}\cup\{e_i\}$), which contradicts the choice of the set T. The resulting contradiction proves the theorem.

We illustrate the proposed theoretical provisions by example.

Possible communication lines are presented in the form of a graph G in figure 3. Next to the edges representing the

communication lines, the reliability of these lines is indicated (for example, the degree of our confidence that there will be no unauthorized access to information during the transmission of information on this line). The main source of information (the point of issue of directives) is points 1, i.e. it is necessary to transfer messages from points 1 to all other points. To determine what channels of communication should be created to allow the reliable transmission of information without violating its confidentiality in the context of this communications system would be the maximum.

Define $G_0 = (V, \varnothing)$, $V = \{1,2,3,4\}$. We list the edges that make up the MakO, specifying them in the order in which they were attached to the created skeleton:

$$(4,6), (3,4), (4,5), (2,5), (1,2).$$

This is the skeleton of maximum reliability; its reliability index is 0,7.

Note that the order of construction of the skeleton could be different. Note also that joining an edge $(3,5)$ instead of $(2,5)$, say, impossible, since the edges $(3,5), (3,4), (4,5)$ form a cycle.

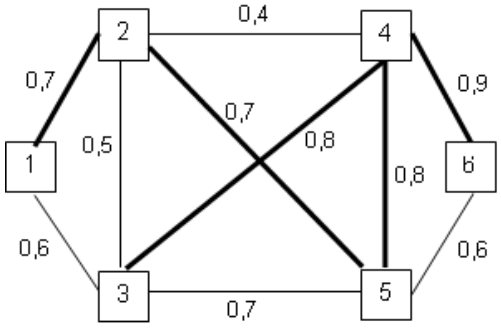


Fig. 3. Graph, which shows all possible communication channels

Let's complicate a little the formulation of the considered problem, we will take into account not only the indicator of the reliability of the communication channel, but also the cost of its creation. Let's complicate a little the formulation of the considered problem, we will take into account not only the indicator of the reliability of the communication channel, but also the cost of its creation. Let each edge of the graph $G = (V, E)$ correlate the degree of truth of the statement «communication channel is reliable» (reliability) and the value of its creation (cost). Let us consider two problems in connection with the above.

1. Build the most reliable skeleton, provided that its cost should not be more *S*.

This problem can be solved by consistently excluding the edges of minimal reliability from the original graph.

The decision starts with the consideration of the initial graph, which is built the skeleton for a minimal cost. Let its cost is less *S* and its reliability is equal r_1. Remove from the original graph communication channels, the reliability of which does not exceed r_1. For the resulting graph again build the skeleton of the minimum cost. The cost and reliability of the resulting skeleton will not be less than the skeleton obtained earlier. Let its reliability is equal r_2. Remove from the original graph communication channels, the reliability of which does not exceed r_2, etc. until we get a skeleton cost more *S* or a graph that does not contain all the vertices of the original.

2. Build a skeleton of the minimum cost, provided that its reliability must be at least *R*.

To solve this problem, we remove from the original graph communication channels, reliability of which is less *R*, and build the skeleton of the minimum cost for the resulting graph. If we get a graph that does not contain all the vertices of the original one, then the formulated problem has no solution.

Conclusion

As a conclusion, we note that the article proposes an easily programmable algorithm for creating a communication system that provides maximum information security, implemented within its framework of information exchange.

References

1. Belousov A.I., Tkachev S.B. Discrete mathematics: Textbook [Diskretnaya matematika: Uchebnik dlya vuzov], Moscow, Bauman Moscow State Technical University, 2006, 744 p.

2. Cormen T.H., Leiserson C.I., Rivest R.L., Stein C. Minimum spanning trees [Minimal`nye ostovnye derev`ya], *In: Introduction to Algorytms. Second Edition [Algoritmy: postroenie i analiz. Vtoroe izdanie]*, Moscow, Williams, 2005, 1296 p.

3. Taha H.A. Operations research: an introduction [Vvedenie v issledovanie operatsiy], Moscow, Williams, 2005, 901 p.

4. Kruskal J.B., Jr. On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. *Proc. of the American Mathematical Society*, 1956, Vol. 7, No. 1, pp.48–50.

5. Chernykh A.K., Vilkov V.B. Management of Transportation Safety at the Organization of Material Maintenance of Forces and Means Emercom of Russia in Emergencies [Upravlenie bezopasnost'yu transportnykh perevozok pri organizatsii material'nogo obespecheniya sil i sredstv MChS Rossii v usloviyakh chrezvychaynoy situatsii], *Fire and Explosion Safety [Pozharovzryvobezopasnost`]*, 2016, Vol. 25, No. 9, pp. 52–59.

6. Vilkov V.B., Chernykh A.K., Garkushev A.Yu., Zaitsev A.I. Algorithm of Finding of The Optimum Route of Promotion of Division of Troops of National Guard [Algoritm poiska optimal'nogo marshruta vydvizheniya podrazdeleniya voysk natsional'noy gvardii], *Bulletin of the Russian Academy of Rocket and Artillery Sciences [Izvestiya Rossiyskoy akademii raketnykh i artilleriyskikh nauk]*, 2017, No. 1 (96), pp. 29–33.

7. Vilkov V.B., Chernykh A.K. Flegontov A.V. Theory and practice of optimization of decisions based on fuzzy sets and fuzzy logic: Monograph. [Teoriya i praktika optimizatsii

resheniy na osnove nechetkikh mnozhestv i nechetkoy logiki: Monografiya], St. Petersburg, Herzen State Pedagogical University of Russia, 2017, 159 p.

8. Vilkov V.B., Shcherbakova O.I., Chernykh A.K., Andreev V.P., Khudyakova T.L., Kazakova S.N. The Choice of an Optimal Methodology for the Retraining Organization of Psychologists Based on the Use of Mathematical Methods, *Espacios*, 2018, Vol. 39, No. 20, p. 16.

9. Kofman A. Introduction to the theory of fuzzy sets [Vvedenie v teoriyu nechetkikh mnozhestv], Moscow, Radio and Communication Publishing House, 1982, 432 p.

10. Leonenkov A.V. Fuzzy modeling in MATLAB and fuzzyTECH [Nechetkoe modelirovanie v srede MATLAB i FuzzyTECH], St. Petersburg, BHV-Peterburg, 2003, 719 p.

11. Orlovsky, S. A. Problems of decision making with fuzzy initial information [Problemy prinyatiya resheniy pri nechetkoy iskhodnoy informatsii], Moscow, Science, 1981, 206 p.

12. Yahyaeva G. E. Fuzzy sets and neural networks: Study guide [Nechetkie mnozhestva i neironnye seti: Uchebnoe posobie], Moscow, BINOM, INTUIT, 2006, 314 p.

13. Zadeh L. Fuzzy sets, *Information and Control*, 1965, No. 8, pp. 338–353.

14. Gladkikh V. P., Shved V. G., Dergachev A.I. Modeling of System of Information Security to the Military Authorities [Osobennosti modelirovaniya sistemy informatsionnoy bezopasnosti v organakh voennogo upravleniya], *National Priorities of Russia. Series 1: Science and Military Security [Natsional'nye prioritety Rossii. Seriya 1: Nauka i voennaya bezopasnost']*, 2015, No. 1 (1). pp. 47–49.

15. Dei Rossi J.A., Heiser R.S., King N.S. A Cost Analysis of Minimum Distance TV Networking for Broadcasting Medical Information, Santa Monica, California, RAND Corp., February 1970, 86 p.

16. Applied graph theory. Lecture No. 7. Trees [Prikladnaya teoriya grafov. Lektsiya 7. Derev`ya]. Available at: http://www.studfiles.ru/preview/3350252 (accessed 08.04.2019).

17. Borisov A. N., Krumberg O. A., Fyodorov I.P. Decision-Making based on fuzzy models: Examples of use [Prinyatie resheniy na osnove nechetkikh modeley: Primery ispol'zovaniya], Riga, Zinatne, 1990, 184 p.

18. Zade L. The Concept of a linguistic variable and its application to making approximate decisions [Ponyatie lingvisticheskoy peremennoy i ego primenenie k prinyatiyu priblizhennykh resheniy], Moscow, Mir Publishing House, 1976, 166 p.

19. Terano T., Asai K., Sugeno M. Applied fuzzy system [Prikladnye nechetkie systemy], Moscow, Mir Publishing House, 1993, 368 p.

20. Stovba S.D. Introduction to fuzzy set theory and fuzzy logic [Vvedenie v teoriyu nechetkikh mnozhestv i nechetkuyu logiku], Vinnytsia, UNIVERSUM-Vinnytsia Publisher, 2001, 756 p.

21. Emelichev V.A., Melnikov O.I., Sarvanov V.I., Tyshkevich R.I. Lectures on graph theory [Lektsii po teorii grafov], Moscow, Science, 1990, 384 p.

Об одном подходе к созданию информационно-безопасных систем связи

к.ф.-м.н. В.Б. Вилков
Военная академия материально-технического обеспечения имени генерала армии А.В. Хрулёва Министерства обороны РФ
Санкт-Петербург, Россия
amirusha@rambler.ru

д.т.н. А.К. Черных
Санкт-Петербургский военный институт войск национальной гвардии Российской Федерации.
Санкт-Петербург, Россия
nataliachernykh@mail.ru

к.воен.н. А.И. Дергачёв
Петербургский государственный университет путей сообщения Императора Александра I
Санкт-Петербург, Россия
d_ader@mail.ru

Аннотация. **Рассматривается подход к созданию систем связи, в максимальной степени обеспечивающих информационную безопасность обрабатываемых в их рамках данных. Предложен оптимальный алгоритм, реализующий указанный подход, который проиллюстрирован примером. Приведены постановки задачи, реализующей создание указанных систем связи, основанные на идеях многокритериальной оптимизации.**

Ключевые слова: **информационная безопасность системы, надежность системы связи, теория нечетких множеств, нечеткая логика, ребра графов, максимальный остов.**

Литература

1. Белоусов А.И., Ткачев С.Б. Дискретная математика : учебник для вузов / под ред. В.С. Зарубина, А.П. Крищенко. — 4-е изд., исправл. — М. : МГТУ им. Н.Э. Баумана, 2006. — 744 с. (Математика в техническом университете; Вып. XIX).

2. Кормен Т.Х. Минимальные остовные деревья / Т.Х. Кормен, Ч.И. Лейзерсон, Р.Л. Ривест, К. Штайн // Алгоритмы: построение и анализ. – 2-е изд. = Introduction to Algorithms. Second Edition: пер. с англ. — М. : Вильямс, 2005. — 1296 с.

3. Таха Х.А. Введение в исследование операций / пер. с англ. и ред. А.А. Минько. —7-е изд. — М. : Вильямс, 2005. — 901 с.

4. Kruskal J.B., Jr. On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. *Proc. of the American Mathematical Society*, 1956, Vol. 7, No. 1, pp.48–50.

5. Черных А.К., Вилков В.Б. Управление безопасностью транспортных перевозок при организации материального обеспечения сил и средств МЧС России в условиях чрезвычайной ситуации // Пожаровзрывобезопасность. — 2016. — Т. 25, № 9. — С. 52–59.

6. Вилков В.Б. Алгоритм поиска оптимального маршрута выдвижения подразделения войск национальной гвардии / В.Б. Вилков, А.К. Черных, А.Ю. Гарькушев, А.И. Зайцев // Известия Российской академии ракетных и артиллерийских наук. – 2017. – № 1 (96). – С. 29–33.

7. Вилков В. Б. Теория и практика оптимизации решений на основе нечетких множеств и нечеткой логики : монография / В.Б. Вилков, А.К. Черных, А.В. Флегонтов. — СПб. : РГПУ им. А.И. Герцена, 2017. — 159 с.

8. Vilkov V.B., Shcherbakova O.I., Chernykh A.K., Andreev V.P., Khudyakova T.L., Kazakova S.N. The Choice of an Optimal Methodology for the Retraining Organization of Psychologists Based on the Use of Mathematical Methods, *Espacios*, 2018, Vol. 39, No. 20, p. 16.

9. Кофман А. Введение в теорию нечетких множеств / пер. с франц. В.Б. Кузьмина ; под ред. С.И. Травкина. — М. : Радио и связь, 1982. — 432 с.

10. Леоненков А.В. Нечеткое моделирование в среде MATLAB и fuzzyTECH. — СПб. : БХВ-Петербург, 2003. — 719 с.

11. Орловский С.А. Проблемы принятия решений при нечеткой исходной информации. — М. : Наука, 1981. — 206 с. (Оптимизация и исследование операций).

12. Яхъяева Г.Э. Нечеткие множества и нейронные сети : учебное пособие. — М. : БИНОМ. Лаборатория знаний, НОУ «ИНТУИТ», 2006. — 314 с. (Основы информационных технологий).

13. Zadeh L. Fuzzy sets, *Information and Control*, 1965, No. 8, pp. 338–353.

14. Особенности моделирования системы информационной безопасности в органах военного управления / В.П. Гладких, В.Г. Дергачев А.И. Швед // Национальные приоритеты России. Сер. 1. Наука и военная безопасность. — 2015. — № 1 (1). — С. 47–49.

15. Dei Rossi J.A., Heiser R.S., King N.S. A Cost Analysis of Minimum Distance TV Networking for Broadcasting Medical Information, Santa Monica, California, RAND Corp., February 1970, 86 p.

16. Прикладная теория графов. Лекция № 7. Деревья. — URL: http://www.studfiles.ru/preview/3350252 (дата обращения 08.04.2019).

17. Борисов А.Н. Принятие решений на основе нечетких моделей. Примеры использования / А.Н. Борисов, О.А. Крумберг, И.П. Федоров. — Рига : Зинатне, 1990. — 184 с.

18. Заде Л. Понятие лингвистической переменной и его применение к принятию приближенных решений / пер. с англ. Н.И. Ринго ; под ред. Н.Н. Моисеева, С.А. Орловского. — М. : Мир, 1976. — 166 с. (Математика: новое в зарубежной науке; Вып. 3).

19. Прикладные нечеткие системы / пер. с япон. Ю.Н. Чернышова; К. Асаи, Д. Ватада, С. Иваи и др.; под ред. Т. Тэрано, К.Асаи, М. Сугэно. — М. : Мир, 1993. — 368 с.

20. Штовба С.Д. Введение в теорию нечетких множеств и нечеткую логику. — Винница : УНИВЕРСУМ-Винница, 2001. — 756 с.

21. Лекции по теории графов / В.А. Емеличев, О.И. Мельников, В.И. Сарванов, Р.И. Тышкевич. — М. : Наука, 1990. — 384 с.

Обзор кодов для помехоустойчивого кодирования

О.А. Турдиев, д.т.н. В.В. Яковлев
Петербургский государственный университет путей сообщения Императора Александра I
Санкт-Петербург, Россия
odiljan.turdiev@mail.ru, jakovlev@pgups.ru

С.В. Клименко
Инженер-программист Dell EMC
Санкт-Петербург, Россия
s.klimenko@live.ru

Аннотация. **Рассматриваются базовые алгоритмы помехоустойчивого кодирования для обеспечения достоверности передаваемых и хранимых данных в вычислительных системах и сетях передачи данных. Подробно описаны принципы работы кодов, таких как бит четности, вертикальный и горизонтальный контроль по четности и код Хэмминга, и сфера их применения.**

Ключевые слова: **четность, помехи, кодирование, декодирование.**

Введение

При передаче сообщения (данных) от источника к приемнику может произойти ошибка (помехи, неисправность оборудования и пр.). Для обнаружения и исправления ошибок применяют помехоустойчивое кодирование, т. е. кодируют сообщение таким образом, чтобы принимающая сторона знала, произошла ошибка или нет, и могла исправить ошибки в случае их возникновения. Применение помехоустойчивого кодирования с исправлением ошибок в современных системах связи является обязательным. Под кодированием понимается добавление к исходной информации дополнительной проверочной информации. Для кодирования на передающей стороне используется кодер, а на принимающей стороне для получения исходного сообщения используют декодер (рис. 1), [1, 2].

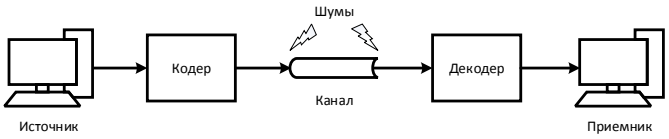


Рис. 1. Прохождение сигнала по каналу связи

При использовании кодирования возникает понятие избыточности кода. Под ним понимается количество проверочной информации в сообщении.

Бит четности

Бит четности (бит паритета) — метод для обнаружения ошибок в передаваемом пакете данных. Позволяет обнаружить одиночную ошибку, без возможности восстановить данные [3, 4].

Этот бит рассчитывается и устанавливается во время отправки данных кодером (рис. 2); после получения данных декодером (рис. 3) он рассчитывается заново и сравнивается с полученным. Для расчета бита четности используется булева функция — сумма по модулю 2 (исключающее «ИЛИ») для всех бит данных. Изменение значения этого бита с 0 на 1 или с 1 на 0 свидетельсвует о наличии ошибки [5].

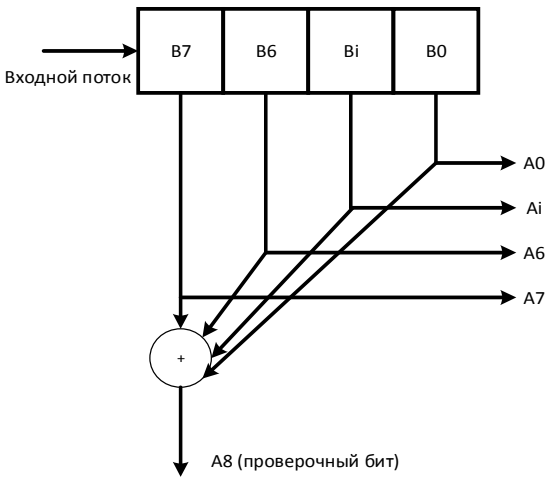


Рис. 2. Структурная схема работы кодера

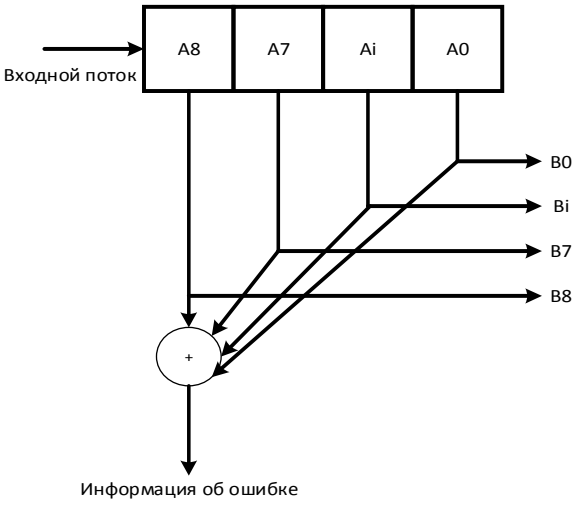


Рис. 3. Структурная схема работы декодера

Многие реализации UART (Universal Asynchronous Receiver-Transmitter — универсальный асинхронный приемопередатчик) имеют возможность автоматически контролировать целостность данных методом контроля битовой четности. Когда эта функция включена, последний бит данных в минимальной посылке («бит четности») контролируется логикой UART и содержит информацию о четности количества единичных бит в этой минимальной посылке.

Вертикальный и горизонтальный контроль по четности

Представляет собой модификацию описанного выше метода. Его отличие состоит в том, что исходные данные рассматриваются в виде матрицы, строки которой составляют байты данных. Контрольный разряд подсчитывается отдельно для каждой строки и для каждого столбца матрицы. Пример построения и проверки метода на основе формирования BCC (Block Check Character – проверка суммы блока) показан на рисунке 4. Метод позволяет обнаружить большую часть двойных ошибок, однако обладает еще большей избыточностью.

На практике метод почти не применяется из-за его большой избыточности и невысокой диагностической способности.

а)

A	B_7	B_6	B_5	B_4	B_3	B_2	B_1
0	0	0	0	0	0	1	0
1	0	1	0	1	0	0	0
0	1	0	0	0	1	1	0
0	0	1	0	0	0	0	0
1	0	1	0	1	1	0	1
0	1	0	0	0	0	0	0
1	1	1	0	0	0	1	1
1	0	0	0	0	0	1	1
1	1	0	0	0	0	0	1

б)

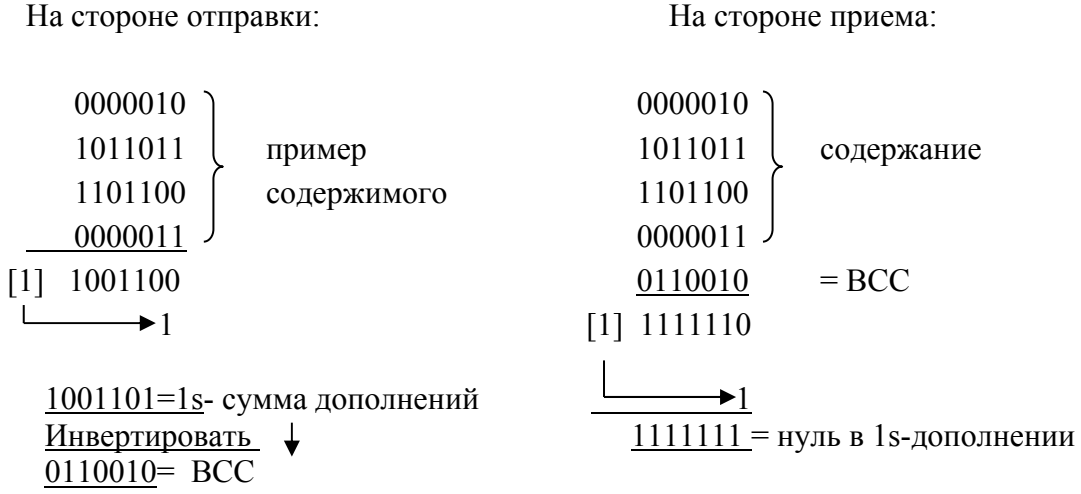


Рис. 4. Метод проверки на основе формирования BCC: *а* – пример построения вертикального и горизонтального контроля по четности; *б* – формирование дополнительного кода суммы

Код Хэмминга

Коды Хэмминга — наиболее известные из семейства самокорректирующихся кодов. Построены они применительно к двоичной системе счисления [6, гл. 5; 7].

Код Хэмминга позволяет закодировать какое-либо информационное сообщение на основе представленного на рисунке 5 алгоритма и после передачи определить, появилась ли какая-либо ошибка в этом сообщении (ввиду помех, неисправности оборудования и пр.), и при возможности восстановить это сообщение [8, 9].

Для каждого числа проверочных символов используется специальная маркировка вида (k, i), где k — количество символов в сообщении, i — количество информационных символов в сообщении. Например, существуют (см. рис. 5) коды (7, 4) (15, 11), (31, 26). Каждый проверочный символ в коде Хэмминга представляет сумму по модулю 2 некоторой подпоследовательности данных.

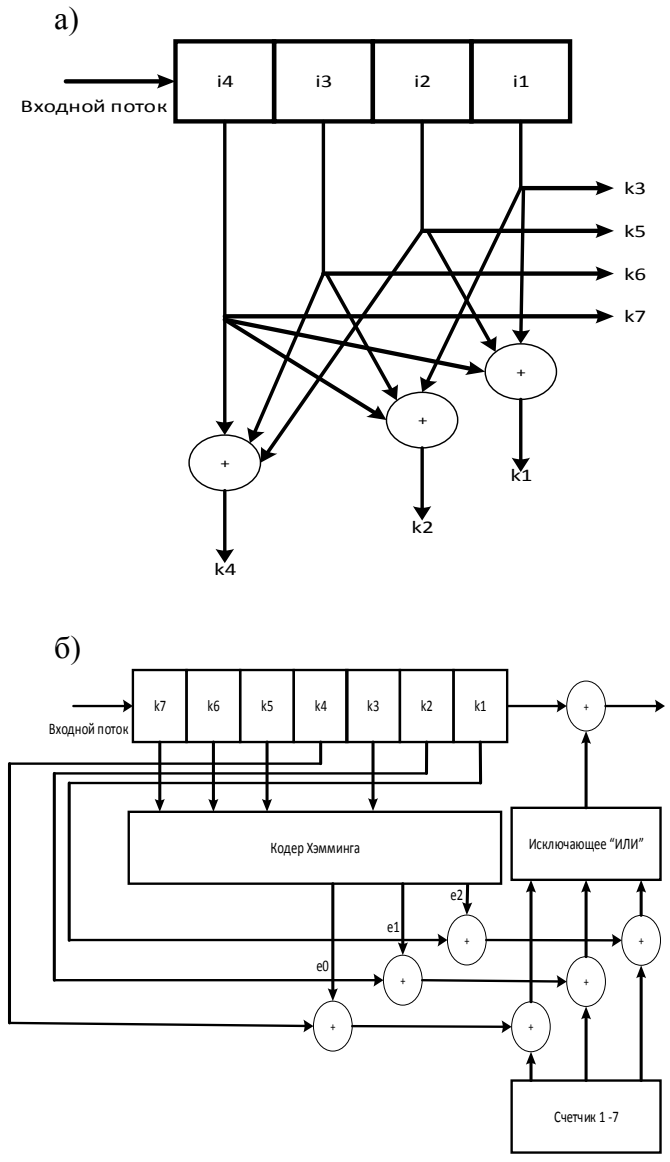


Рис. 5. Структурная схема работы декодера

Код Хэмминга используется в некоторых прикладных программах в области хранения данных, например в RAID-2. Кроме того, метод Хэмминга давно применяется в памяти типа ECC (error-correcting code memory — память с коррекцией ошибок), позволяет «на лету» исправлять однократные и обнаруживать двукратные ошибки [10].

Заключение

В статье рассмотрены простые методы обнаружения и исправления ошибок. Представлены структурные схемы работы кодеров, декодеров для каждого из них, отмечены принципиальные отличия методов друг от друга, а также определена сфера применения.

Литература

1. Ромащенко А.Е. Заметки по теории кодирования / А.Е. Ромащенко, А.Ю. Румянцев, А. Шень. — 2-е изд., испр. и доп. — М. : МЦНМО, 2017. — 88 с.
2. Яковлев В. В. Оценка влияния помех на производительность протоколов канального уровня / В. В. Яковлев, Ф. И. Кушназаров // Известия Петербургского университета путей сообщения. – 2015. – № 1 (42). – С. 133–138.
3. Halsall, F. Computer Networking and the Internet, 5th edition, Addison-Wesley, 2005, 832 p.
4. Олифер В. Г. Компьютерные сети. Принципы, технологии, протоколы : учебник для вузов / В. Г. Олифер, Н. А. Олифер. — 3-е изд. — СПб. : Питер, 2006. — 958 с.
5. Halsall, F. Data Communications, Computer Networks and Open Systems, 4th edition, Addison-Wesley, 1996, 928 p.
6. Уоррен Г., мл. Алгоритмические трюки для программистов = Hacker`s Delight / пер. с англ. — 2-е изд., испр. — М. : Вильямс, 2007. — 289 с.
7. Питерсон У., Уэлдон Е. Коды, исправляющие ошибки / пер. с англ. ; под ред. Р. Л. Добрушина и С. И. Самойленко. — М. : Мир, 1976. — 594 с.
8. Радиотехнические системы передачи информации : учебное пособие для радиотехн. спец. вузов / П.И. Пенин, Л.И. Филиппов. — М. : Радио и связь, 1984. — 256 с.
9. Блейхут Р.Э. Теория и практика кодов, контролирующих ошибки / пер. с англ. И. И. Грушко, В. М. Блиновского ; под ред. К. Ш. Зигангирова. — М. : Мир, 1986. — 576 с.
10. Галлагер Р. Теория информации и надежная связь / пер. с англ.; под ред. М.С. Пинскера и Б.С. Цыбакова. — М. : Советское радио, 1974. — 720 с.

Overview of Codes for Error-Correcting Coding

O.A. Turdiyev, Grand PhD V.V. Yakovlev
Emperor Alexander I Petersburg State Transport
University St. Petersburg, Russia
odiljan.turdiev@mail.ru, jakovlev@pgups.ru

S.V. Klimenko
Engineer software Dell EMC
St. Petersburg, Russia
s.klimenko@live.ru

Abstract. **The article discusses the basic error-correcting coding algorithms to ensure the reliability of transmitted and stored data in computing systems and data networks. The principles of operation of codes (such as: parity bit, vertical and horizontal parity check and Hamming code) and scope are described in detail.**

Keywords: **parity, interference, encoding, decoding.**

REFERENCES

1. Romashchenko A. Ye., Rumyantsev A.Yu., Shen A. Notes on coding theory [Zametki po teorii kodirovaniya], Moscow, Moscow Center for Continuous Mathematical Education, 2017, 88 p.
2. Yakovlev V.V., Kushnazarov F.I. Evaluation of the Effect of Interferences on Link-Layer Protocol Performance [Otsenka vliyaniya pomekh na proizvoditel'nost' protokolov kanal'nogo urovnya], *Proc. of Petersburg Transport University [Izvestiya Peterburgskogo universiteta putey soobshcheniya]*, 2015, No. 1 (42), pp. 133–138.
3. Halsall, F. Computer Networking and the Internet, 5th edition, Addison-Wesley, 2005, 832 p.
4. Olifer V.G., Olifer N.A. Computer network. Principles, technologies, protocols: Textbook [Komp'yuternye seti. Printsipy, tekhnologii, protokoly: Uchebnik dlya vuzov], Saint Petersburg, Peter, 2006, 958 p.
5. Halsall, F. Data Communications, Computer Networks and Open Systems, 4th edition, Addison-Wesley, 1996, 928 p.
6. Warren H.S., Jr. Hacker`s Delight [Algoritmicheskie tryuki dlya programmistov], Moscow, Williams, 2007, 289 p.
7. Peterson W., Weldon E., Jr. Error-Correcting Codes [Kody, ispravlyayushchie oshibki], Moscow, Mir Publishers, 1976, 594 p.
8. Penin P.I., Filippov L.I. Radio transmission system information: Study guide [Radiotekhnicheskie sistemy peredachi informatsii: Uchebnoe posobie], Moscow, Radio and Communication Publishers, 1984, 256 p.
9. Blahut R. E. Theory and practice of error control codes [Teoriya i praktika kodov, kontroliruyushchikh oshibki], Moscow, Mir Publishers, 1986, 576 p.
10. Gallager R. G. Information Theory and Reliable Communication [Teoriya informatsii i nadezhnaya svyaz'], Moscow, Soviet Radio, 1974, 720 p.

Оптимизация приложений для дальтоников

студент магистратуры Н.К. Уваров
Петербургский государственный университет путей сообщения Императора Александра I
Санкт-Петербург, Россия
<nick553@mail.ru>

Аннотация. **Рассматривается проблематика адаптации приложений для людей с отклонениями в цветоощущении. Приводится разбор источника проблемы, рассматриваются возможные методы ее решения, а также ряд примеров реализации этих методов. Один из самых простых способов адаптировать приложение для дальтоников — использовать символы, текст или различные изображения вкупе с цветом. Примером такого подхода служит приложение Trello — органайзер, реализованный как веб-сервис. Еще два способа обеспечения доступности приложения для дальтоников: использование особых палитр цветов и просмотр всего дизайна в монохроме.**

Ключевые слова: **дальтонизм, приложение, цвет в ЭВМ, компьютерная графика.**

Введение

Дальтонизм, или цветовая слепота, — это нарушение восприятия цветов, вызванное расстройством цветного зрения. Следует отметить, что это одно из распространенных нарушений зрения: около 8 % мужчин и 1 % женщин имеют дальтонизм какой-либо степени [1]. Это означает, что почти каждый десятый человек, пользующийся тем или иным программным продуктом (приложением) будет иметь трудности с правильным пониманием (обработкой) визуальных данных. Если работа приложения зависит от восприятия цвета пользователем, то это может сказаться на пользовательском опыте, пусть даже и одного из десяти человек, т. к. именно сбалансированная инфографика предполагает грамотное использование визуального представления (например, диаграмм, графиков, иконок, изображений), соответствующий выбор цветов и шрифтов и др.

Так как же нам настроить приложение? Как исправить проблемы? Как можно предотвратить эти проблемы?

Представление цвета в ЭВМ

При построении модели цвета в ЭВМ опираются в первую очередь на то, как работает человеческий глаз. Его можно представить как оптический прибор со своим «объективом» — хрусталиком — и светочувствительным элементом — сетчаткой. С помощью глаза можно различить не все электромагнитные волны, а только те, длина которых находится в диапазоне 400...700 нм. Хрусталик проецирует видимое изображение на сетчатку, покрытую светочувствительными рецепторами — палочками и колбочками. Строение глаза можно увидеть на рисунке 1.

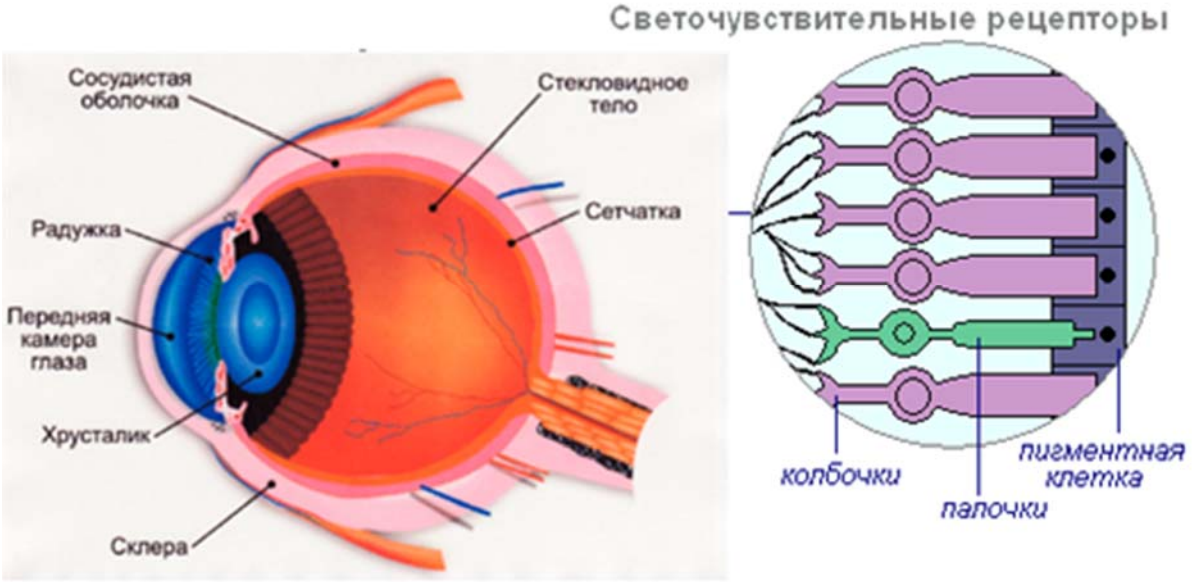


Рис. 1. Строение человеческого глаза

Рецепторы содержат несколько типов цветочувствительных пигментов белкового происхождения. Так, например, в колбочках содержится йодопсин (общее название зрительных пигментов, содержащихся в колбочках сетчатки). В состав йодопсина входят три пигмента, один из них — хлоролаб, специфический фоточувствительный пигмент. Хлоролаб получил свое название в связи со специфическим спектром поглощения в видимой области спектра, с максимальной чувствительностью к области, соответствующей желто-зеленой части спектра (максимум около 534–545 нм). Второй — эритролаб, максимальная чувствительность к области, соответствующей желто-красной части спектра (около 564–580 нм). Третий пигмент — цианолаб, с максимальной чувствительностью к сине-фиолетовой части спектра (420–440 нм) [2].

Палочки, расположенные в основном на краю сетчатки, обладают крайне высокой светочувствительностью, но при этом слабо различают длину волны. Колбочки, расположенные в основном в центре сетчатки, обладают низкой светочувствительностью и узким диапазоном воспринимаемых длин волн (рис. 2).

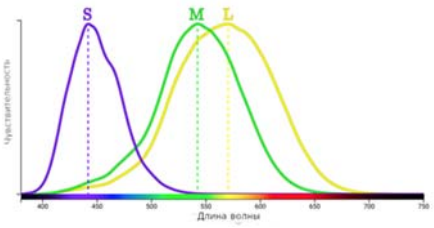


Рис. 2. Чувствительность колбочек к длинам волн

Таким образом, цвет можно задавать как сумму трех базовых цветов — красного, зеленого и синего. Такая модель получила название RGB. Эту модель можно представить в виде куба (рис. 3), в котором каждый цвет представляется трехмерным вектором, компоненты которого определяют доли красного, зеленого и синего цветов, смешивая которые можно получить все остальные цвета и оттенки. В этой модели черный цвет представлен вектором (0, 0, 0), белый цвет — (1, 1, 1) [3].

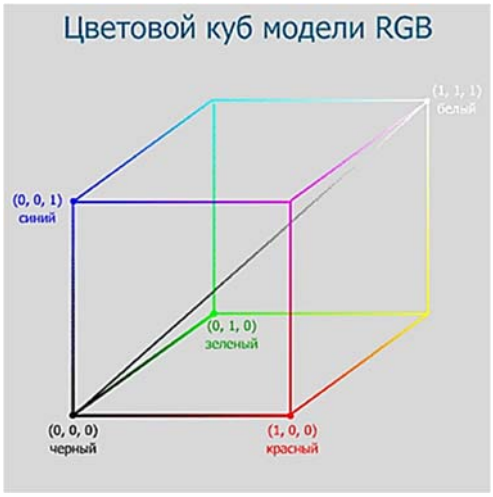


Рис. 3. Куб представления цвета в RGB

Несмотря на всеобщее использование модели RGB, она не является стандартом. Стандартом является модель CIE XYZ, в которой каждый видимый человеком цвет однозначно представляется с помощью тройки неотрицательных чисел (X, Y, Z). Для получения этих чисел используются формулы

$$X = \int I(\lambda)\, k_x(\lambda), \qquad Y = \int I(\lambda)\, k_y(\lambda),$$

$$Z = \int I(\lambda)\, k_z(\lambda), \qquad (1)$$

где $k_x(\lambda), k_y(\lambda), k_z(\lambda)$ — определенные стандартом базовые функции (рис. 4).

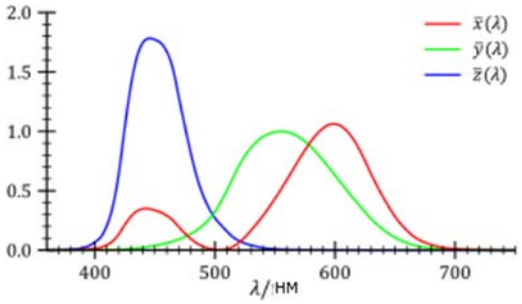


Рис. 4. График функций $k_x(\lambda), k_y(\lambda), k_z(\lambda)$

В представленных моделях все три координаты несут в себе информацию о цвете. Тем не менее существуют модели с явным делением координат на яркость и хроматические координаты. Одна из таких моделей – Yxy, которая определяется следующими формулами:

$$x = \frac{X}{X+Y+Z},\ y = \frac{Y}{X+Y+Z}. \qquad (2)$$

Хроматические координаты x и y полностью определяют цвет без учета его яркости.

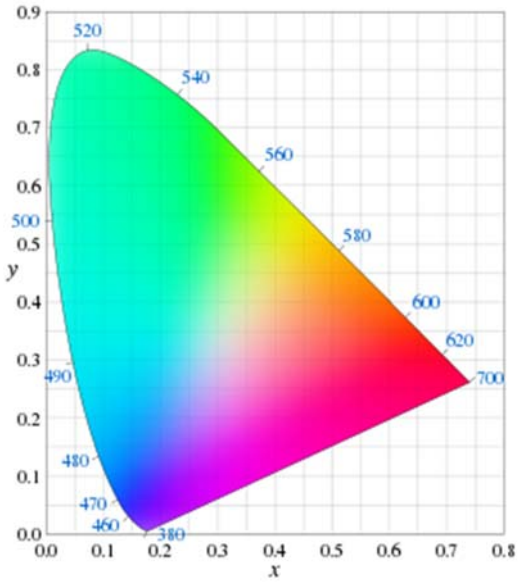


Рис. 5. Хроматическая диаграмма

На рисунке 5 представлена хроматическая диаграмма, которая содержит хроматические координаты всех видимых человеком цветов [4]. На этом изображении дуга соответствует чистым спектральным цветам, а отрезок — цветам, получающимся при смешивании красного и синего цветов.

Если взять два цвета, то им будут соответствовать две точки на диаграмме. Все множество цветов, получаемое смешиванием этих двух цветов, будет выражено отрезком между этими двумя точками. Соответственно при трех цветах множество будет выражено треугольником.

Вывод: никакое конечное количество цветов не может дать при смешивании все видимые человеком цвета.

Несмотря на то, что цветовое пространство CIE XYZ является стандартом, на практике почти для всех светоизлучающих устройств используют модель RGB, поскольку она проста и удобна; даже несмотря на то, что она не может дать все видимые цвета, ее почти всегда бывает достаточно.

Существует еще несколько моделей представления:

- CMY (Cyan, Magenta, Yellow) — является обратной модели RGB и используется при печати на бумаге.
- HSV (Hue, Saturation, Value) — выявляет цвет по яркости, насыщенности и тону.
- HSL (Hue, Saturation, Lightness) — цветовая модель, в которой цветовыми координатами являются тон, насыщенность и светлота [5].

Дальтонизм

Дальтонизм (цветовая слепота) — наследственная, реже приобретенная особенность зрения, выражающаяся в неспособности различать один или несколько цветов и оттенков. Названа в честь Джона Дальтона, который впервые, в 1794 году, дал широкодоступное описание одного из видов цветовой слепоты на основании собственных ощущений.

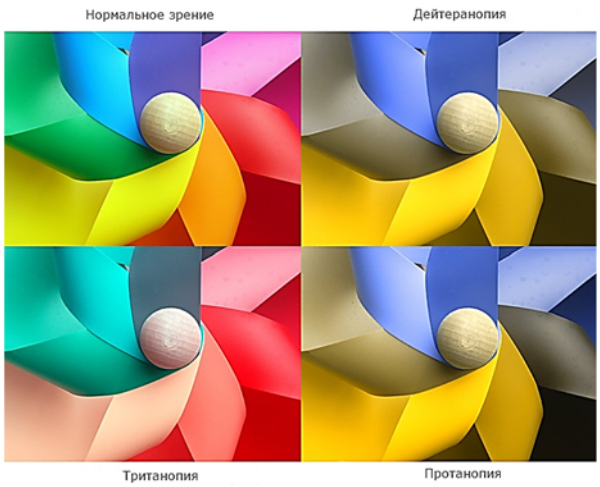


Рис. 6. Виды дальтонизма

Люди с нормальным цветным зрением имеют в рецепторах все три пигмента (эритролаб, хлоролаб и цианолаб) в необходимом количестве. Их называют трихроматами (от слова *хрома* — цвет).

Существует несколько типов дальтонизма:

- монохроматизм, имеется либо один пигмент, либо полное их отсутствие;
- дихроматизм, имеются два пигмента, третий полностью отсутствует;
- аномальный трихроматизм, имеются все три пигмента, но смещен пик чувствительности для одного из них, в результате чего получается меньший спектр цветов.

Дихроматы и аномальные трихроматы также делятся на три типа (рис. 6):

- тританопия — отсутствие/неисправность колбочек с цианолабом (синий);
- дейтеранопия — отсутствие/неисправность колбочек с хлоролабом (зеленый);
- протанопия — отсутствие/неисправность колбочек с эритролабом (красный) [6].

В результате того, что многие цвета являются сочетаниями других цветов, можно заметить, что дальтонизм влияет на весь цветовой спектр.

На рисунке 7 представлены хроматические диаграммы с линиями спутывания. Все цвета в направлении линий являются трудноразличимыми. Место, где они сходятся, называется копунктуальной точкой. Для трех разных типов дальтонизма — три разные копунктуальные точки.

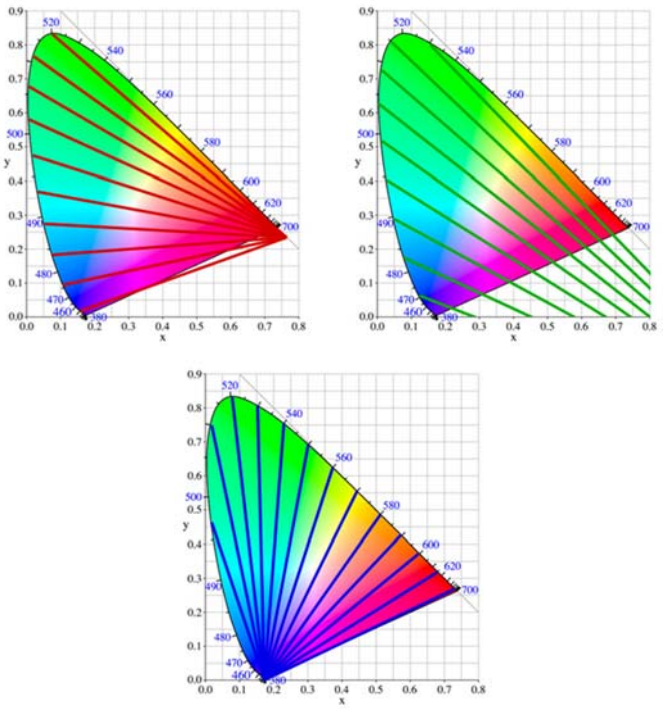


Рис. 7. Хроматические диаграммы с линиями спутывания для разных видов дальтонизма

Адаптация приложений для дальтоников

Один из самых простых способов адаптировать приложение для дальтоников — использовать символы, текст или различные изображения вкупе с цветом. Примером такого подхода может служить приложение Trello — органайзер, реализованный как веб-сервис [7]. На рисунке 8 видно, как эта программа меняет вкладки в зависимости от включенного режима для дальтоников.

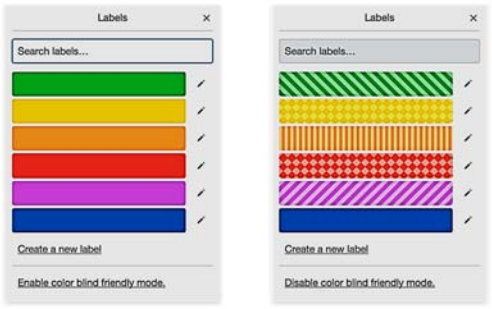


Рис. 8. Интерфейс приложения Trello

Следующий пример — игра Two Dots, которая основана на соединении точек одного цвета [8]. С нарушением цветоощущения в эту игру совершенно невозможно играть, тем не менее разработчики реализовали режим, который добавляет еще один признак для классификации в виде рисунка или символа на цвете (рис. 9).

Рис. 9. Интерфейс игры Two Dots

Еще одним способом обеспечения доступности приложения для дальтоников является использование особых палитр цветов. На рисунке 10 представлена палитра из 15 цветов, сделанная специально для случаев дейтеранопии. Она сделана с помощью модели представления HSL, а не RGB. HSL особенно хорошо подходит для создания палитр для дальтоников, так как они больше ориентируются на яркость (Luminosity), чем на сам цвет (Hue).

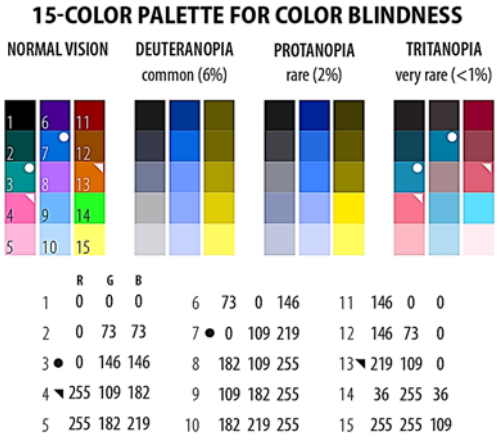


Рис. 10. Палитра цветов для дейтеранопии

Данная палитра хороша для дейтеранопов, но не подходит для нормального зрения. Обратная ситуация будет, если строить палитру только для обычного зрения.

Чтобы сделать приложение доступным для всех типов цветоощущения, необходимо создать разные палитры для каждого из них.

Множество компаний используют этот способ. Например, в Android (рис. 11) и iOS уже в самой ОС встроены средства для изменения палитры цветов [9].

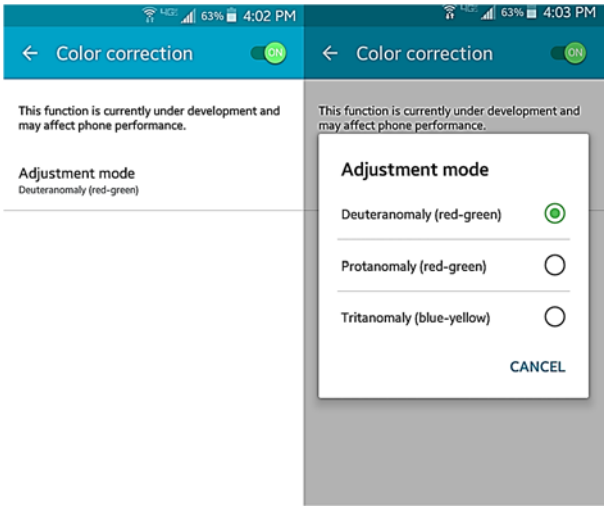


Рис. 11. Режимы для дальтоников в ОС Android

Одной из самых удачных попыток имплементации данного способа является видеоигра Destiny 2 [10]. В игре, где цвет играет очень важную роль, разработчики точно определили, какие цвета необходимы для каждой палитры, тем самым обеспечив одинаковый опыт для всех пользователей. На рисунке 12 можно увидеть окно настройки палитры и примеры цветов.

Рис. 12. Настройки режима для дальтоников в Destiny 2

Для проверки выбранных решений можно воспользоваться специальными симуляторами дальтонизма, которые активируют фильтр, имитирующий зрение человека с проблемным цветоощущением. С помощью этих фильтров можно проверить правильность выбранных цветов.

Еще одним способом является просмотр всего дизайна в монохроме. Этот тест очень полезен для выявления сливающихся оттенков, так как он позволит увидеть, какие цвета одинаковой теплоты имеют схожие оттенки.

Заключение

Очень важно при создании приложения сделать его доступным для людей с отклонениями, которые при нормальных условиях не смогли бы им пользоваться.

Это не только поднимет престиж данного приложения на рынке, но и привлечет новую часть аудитории.

Литература

1. Color Blindness. Prevent Blindness.– URL: http://www.preventblindness.org/color-blindness (дата обращения 10.01.2019)

2. Самаль И.Н. Анатомия, физиология и патология органа зрения : учебное пособие. — Псков : ПГПУ им. С.М. Кирова, 2004. — 164 с.

3. Боресков А.В. Компьютерная графика : учебник и практикум для прикладного бакалавриата. / А.В. Боресков, Е.В. Шикин. — М. : Юрайт, 2016. — 219 с. (Бакалавр. Прикладной курс)

4. Jackson, K.G., Townsend, G.B. TV & Video Engineer's Reference Book, Oxford, Butterworth-Heinemann Ltd., 1991, 946 p.

5. Различные цветовые модели и их использование. – URL: http://tm.spbstu.ru/Различные_цветовые_модели_и_их_использование (дата обращения 10.01.2019).

6. McIntyre, D. Colour Blindness: Causes and Effects. Chester, UK, Dalton Publishing, 2002, 112 p.

7. Trello. — URL: http://trello.com (дата обращения 10.01.2019).

8. Two Dots. Free puzzle game for iOS and Android. URL: http://www.dots.co/twodots (дата обращения 10.01.2019).

9. Dobie, A. Android L includes new display modes for color blind users, *Androidcentral*, 27 Jun 2014. – URL: http://www.androidcentral.com/android-l-includes-new-display-modes-color-blind-users (дата обращения 10.01.2019).

10. Game accessibility guidelines. Destiny colorblind modes. – URL: http://gameaccessibilityguidelines.com/destiny-colorblind-modes (дата обращения 10.01.2019).

Adapting Applications for Colorblind Users

Graduate Student N.K. Uvarov
Emperor Alexander I St. Petersburg State Transport University
St. Petersburg, Russia
nick553@mail.ru

Abstract. **The problems of adapting applications for people with color perception are considered. An analysis of the source of the problem is given, possible methods for solving it are considered, and a number of examples of the implementation of these methods are given. One of the easiest ways to adapt an application for color blindness is to use symbols, text, or various images along with color. An example of this approach is the application Trello — organizer, implemented as a web service. Two more ways to ensure the availability of applications for distance tonics: the use of special color palettes and view the entire design in monochrome.**

Keywords: **colorblind, applications, color in computers, computer graphics.**

REFERENCES

1. Color Blindness. Prevent Blindness. Available at: http://www.preventblindness.org/color-blindness (accessed 10 Jan 2019).

2. Samal I.N. Anatomy, physiology and pathology of the organ of vision [Anatomiya, fiziologiya i patologiya organa zreniya], Pskov, Pskov State Pedagogical University, 2004, 164 p.

3. Boreskov A.V., Shikin Y.V. Computer graphics: textbook and workshop [Komp'yuternaya grafika: uchebnik i praktikum dlya prikladnogo bakalavriata], Moscow, Urait Publishing House, 2016, 219 p.

4. Jackson, K.G., Townsend, G.B. TV & Video Engineer's Reference Book, Oxford, Butterworth-Heinemann Ltd., 1991, 946 p.

5. Different color models and their use [Razlichnye tsvetovye modeli i ikh ispol'zovanie] Available at: http://tm.spbstu.ru/Различные_цветовые_модели_и_их_использование (accessed 10 Jan 2019).

6. McIntyre, D. Colour Blindness: Causes and Effects. Chester, UK, Dalton Publishing, 2002, 112 p.

7. Trello. Available at: http://trello.com (accessed 10 Jan 2019).

8. Two Dots. Free puzzle game for iOS and Android. Available at: http://www.dots.co/twodots (accessed 10 Jan 2019).

9. Dobie, A. Android L includes new display modes for color blind users, *Androidcentral*, 27 Jun 2014. Available at: http://www.androidcentral.com/android-l-includes-new-display-modes-color-blind-users (accessed 10 Jan 2019).

10. Game accessibility guidelines. Destiny colorblind modes. Available at: http://gameaccessibilityguidelines.com/destiny-colorblind-modes/ (accessed 10 Jan 2019).

Инструмент управления тестовыми данными

студент магистратуры Д.Е. Кунгуров, студент магистратуры М.В. Шульга
Петербургский государственный университет путей сообщения Императора Александра I
Санкт-Петербург, Россия
decokungurov@mail.ru, maxvik2507@gmail.com

Аннотация. **При разработке программного обеспечения тестирование занимает одно из ключевых мест и является необходимым для обеспечения наилучшего качества выпускаемого продукта. Тестовые данные напрямую влияют на успех в тестировании и являются важным артефактом. На сегодняшний день существует небольшое количество инструментов для работы с тестовыми данными, которые в свою очередь являются платными либо не подходят полностью или частично к решению поставленных задач, что принуждает инженеров к разработке собственных решений или к интеграции с существующими. Целью данной статьи является демонстрация инструмента управления тестовыми данными.**

Ключевые слова: **тестовые данные, автоматизированные тесты, управление тестовыми данными, качество программного обеспечения, инструмент автоматизированного тестирования.**

Введение

Тестирование программного обеспечения — процесс анализа программного средства и сопутствующей документации с целью выявления дефектов и повышения качества продукта [1, с. 8].

Автоматизированное тестирование — это отдельная дисциплина искусства тестирования. Значительная часть эффективности работы отдела тестирования зависит от того, какие задачи отданы для автоматизации и как эта автоматизация была осуществлена. Автоматизация может как принести огромное облегчение всем тестировщикам, так и завалить работу всего отдела и отложить релиз, премию, отпуск и другие приятные вещи [2, с. 166].

При проектировании тестов наиболее ресурсоемкая часть — это подготовка тестовых данных. Они зависят от типа и цели тестирования, стадии разработки проекта и т. д. Тестовые данные сами по себе не являются артефактом, но заметно влияют на успех или неудачу теста. Тестирование не может быть выполнено без данных для тестов, а именно:

- входных данных для создания условия;
- выходных данных для оценки требования;
- вспомогательных данных (как предварительное условие для теста).

Поэтому необходимо иметь решение для управления тестовыми данными и их хранения.

Анализ и идея

На сегодняшний день существует небольшое количество инструментов для работы с тестовыми данными, они дорого стоят, сложны в интеграции с проектами или вообще не подходят для задач, которые вам необходимо решить. Инженеры по автоматизированному тестированию разрабатывают свои инструменты для работы с тестовыми данными или подходят к данному вопросу нецелесообразно и копируют код из предыдущих проектов, что несет свои трудности и проблемы.

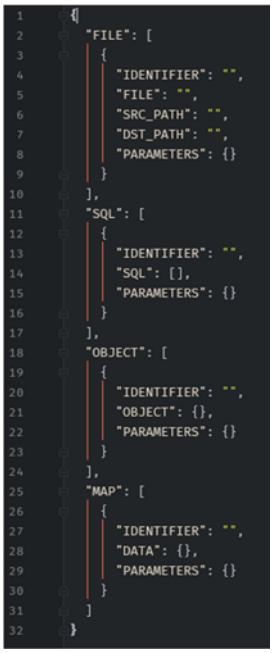
Исходя из сказанного, можно поставить целью создание простого в использовании, гибкого инструмента для эффективного управления тестовыми данными, задачи которого будут состоять в создании среды для работы с тестовыми данными, доставке тестовых данных до тестовых методов и предоставлении методов для работы с ними. Аналогов данного инструмента на просторах интернета найдено не было.

Были проанализированы разные проекты автоматизированного тестирования, которые по своей структуре чем-то схожи и имеют такие источники тестовых данных, как тестовый файл, данные в базе данных или тестовые данные непосредственно в коде и др. Как говорилось выше, тестовые данные можно разделить. Таким образом, при создании инструмента эти факторы нужно учитывать.

Идея инструмента лежит в шаблонах, где будет непосредственно храниться метаинформация и/или тестовые данные разного рода, необходимые для прохождения тестов. Разработанные шаблоны, первый из которых предназначен для зависимых тестовых данных, участвующих в подготовке необходимых условий перед тестом (рис. 1), второй — для тестовых данных, участвующих в тесте (рис. 2), имеют гибкую структуру и хранятся в файлах формата *.json*.

Под гибкой структурой понимается следующее: можно удалять ненужные узлы в файле или добавлять новые, удалять или добавлять значения в узлах; также наименования являются регистронезависимыми. Данные шаблоны преобразуются, и на выходе пользователь получает объекты, которые предоставляют набор различных методов для работы с тестовыми данными.

Данный инструмент можно конфигурировать с помощью аннотаций. Аннотация включает в себя путь до файла с тестовыми данными, тип ожидаемых тестовых данных, тип входных тестовых данных, наименование группы из файла. Последние три являются необязательными и могут быть опущены.



```
{
  "FILE": [
    {
      "IDENTIFIER": "",
      "FILE": "",
      "SRC_PATH": "",
      "DST_PATH": "",
      "PARAMETERS": {}
    }
  ],
  "SQL": [
    {
      "IDENTIFIER": "",
      "SQL": [],
      "PARAMETERS": {}
    }
  ],
  "OBJECT": [
    {
      "IDENTIFIER": "",
      "OBJECT": {},
      "PARAMETERS": {}
    }
  ],
  "MAP": [
    {
      "IDENTIFIER": "",
      "DATA": {},
      "PARAMETERS": {}
    }
  ]
}
```

Рис. 1. Шаблон зависимых тестовых данных

```
{
  "": [
    {
      "CONFIGURATION": {
        "IDENTIFIER": "",
        "DESCRIPTION": "",
        "PARAMETERS": {}
      },
      "SQL": [
        {
          "IDENTIFIER": "",
          "SQL-QUERIES": {},
          "PARAMETERS": {}
        }
      ],
      "EXPECTED": [
        {
          "ENVIRONMENT": "",
          "IDENTIFIER": "",
          "EXPECTED": [{}]
        }
      ],
      "DATASETS": [
        {
          "IDENTIFIER": "",
          "DATASETS": [{}]
        }
      ]
    }
  ]
}
```

Рис. 2. Шаблон тестовых данных

Системные требования

Для работы с инструментом необходимы следующие требования:

- Java версии 8 и выше;
- TestNG;
- Apache Maven.

TestNG — это тестовый фреймворк, вдохновленный JUnit и NUnit, но вводящий некоторые новые функции, которые делают его более мощным и легким в использовании [3].

Провайдер данных (data provider) — один из способов передачи параметров в тестовые методы. Data Provider — это метод в вашем классе, который возвращает массив массивов объектов. Этот метод аннотирован @DataProvider [4].

Apache Maven — это инструмент управления и понимания программного обеспечения. Основываясь на концепции объектной модели проекта (POM), Maven может управлять сборкой проекта, составлением отчетов и документацией из центральной части информации [5].

Быстрый старт

Рассмотрим пример работы с данным инструментом. Пользователю необходимо проделать три простых шага.

Для начала установим инструмент, для этого нужно добавить зависимость в *pom.xml* (рис. 3).

```
<dependency>
    <groupId>ru.mts.qa.tools</groupId>
    <artifactId>testing-data-manipulation-tool</artifactId>
    <version>1.2.0</version>
</dependency>
```

Рис. 3. Добавление зависимости

На первом шаге пользователь берет соответствующий шаблон и вносит всю необходимую информацию (подготовка тестовых данных). Данный файл будет иметь название *testing-data.json* (рис. 4).

```
{
  "CONTEXT": [
    {
      "CONFIGURATION": {
        "IDENTIFIER": "EXAMPLE",
        "DESCRIPTION": "ПРИМЕР ТОЛЬКО С CONFIGURATION",
        "PARAMETERS": {
          "SOMETHING": "SOMETHING"
        }
      }
    }
  ]
}
```

Рис. 4. Файл *testing-data.json*

На втором шаге пользователь создает провайдер данных, используя класс *TDMTTestCaseContext* (рис. 5).

```
class DataProvider {

    @org.testng.annotations.DataProvider
    public Iterator<Object[]> getTestCaseContext(Method method) {
        return TDMTTestCaseContext.getDataProvider(method).iterator();
    }
}
```

Рис. 5. Создание провайдера данных (data provider)

На третьем шаге пользователь создает тестовый метод, где, используя аннотацию *TDMTConfiguration* для конфигурации, указывает путь к файлу с тестовыми данными (рис. 6). Объект *context* класса *TDMTTestCaseContext<Map, Map>* содержит набор различных методов для централизованного и эффективного управления тестовыми данными.

```
class ExampleTest {

    @TDMTConfiguration(datapath = "testing-data.json")
    @Test(dataProviderClass = DataProvider.class, dataProvider = "getTestCaseContext")
    public void test(TDMTTestCaseContext<Map, Map> context) {
        <...>
        String description = context.getConfiguration().getDescription();
        <...>
    }
}
```

Рис. 6. Использование инструмента в тестовом методе

Заключение

Разработанный инструмент прост в использовании и интеграции в проекты, имеет гибкую структуру, централизованное и эффективное управление тестовыми данными. Кроме того, он позволяет повысить читаемость кода, сократить значительное количество человеко-часов на напи-

сание кода по работе с тестовыми данными и является открытым проектом, что делает его доступным любому желающему редактировать или расширить функциональность данного инструмента для своих задач.

Литература

1. Куликов С. С. Тестирование программного обеспечения. Базовый курс. — 2-е изд. — Минск : Четыре четверти, 2017. — 312 с.

2. Савин Р. Тестирование Дот Ком, или Пособие по жестокому обращению с багами в интернет-стартапах. — М. : Дело, 2007. — 312 с.

3. TestNG Overview // Tutorialspoint. — URL: http://www.tutorialspoint.com/testng/testng_overview.htm (дата обращения 06.05.2019).

4. TestNG. URL: http://testng.org/doc/documentation-main.html (дата обращения 06.05.2019).

5. Maven Documentation. — URL: http://maven.apache.org/guides/index.html (дата обращения 06.05.2019).

Testing Data Management Tool

Graduate Student D.Ye. Kungurov, Graduate Student M.V. Shulga
Emperor Alexander I Petersburg State Transport University
Saint Petersburg, Russia
decokungurov@mail.ru, maxvik2507@gmail.com

Abstract. **When developing software, testing takes one of the key places and is necessary to ensure the best quality of the manufactured product. Test data directly affects test success and is an important artifact. Today there are a small number of tools for working with test data, which, in turn, are paid or do not fully or partially fit the solution of the problems that have been set, which leads engineers to develop their own solutions or integrate with existing ones. The purpose of this article is to demonstrate the test data management tool.**

Keywords: **test data, automated tests, test data management, software quality assurance, automated testing tool.**

References

1. Kulikov S.S. Software testing. Basic course [Testirovanie programmnogo obespecheniya. Bazovyy kurs], 2nd edition, Minsk, Four Fourths Publishing House, 2017, 312 p.
2. Savin R. Testing Dot Com, or A Guide to the Abuse of Bugs in Internet Startups [Testirovanie Dot Kom, ili Posobie po zhestokomu obrashcheniyu s bagami v internet-startapakh], Moscow, Delo Publ. House, 2007, 312 p.
3. TestNG Overview. *Tutorialspoint.* Available at: http://www.tutorialspoint.com/testng/testng_overview.htm (accessed 06 May 2019).
4. TestNG. Available at: http://testng.org/doc/documentation-main.html (accessed 06 May 2019).
5. Maven Documentation. Available at: http://maven.apache.org/guides/index.html (accessed 06 May 2019).

Обеспечение безопасности на железнодорожных переездах посредством использования системы GPS/ГЛОНАСС

студент Т.И. Васьков, студент Е.А. Михайленко, к.воен.н. Р.Г. Гильванов
Петербургский государственный университет путей сообщения Императора Александра I
Санкт-Петербург, Россия
timoff.pk@yandex.ru, mihaylenko.evgeniy77@gmail.com, rinatgilvanov@mail.ru

Аннотация. **Основной целью обеспечения безопасности на железнодорожных переездах является снижение дорожно-транспортных происшествий за счет нанесения на навигационную карту навигаторов дополнительных информационных знаков (объектов), отображающих нерегулируемые и регулируемые железнодорожные переезды. В этом случае водитель, двигающийся по маршруту, будет дополнительно получать звуковую и голосовую информацию о приближении к железнодорожному переезду, о типе переезда, его состоянии, разрешенной скорости движения и реальной скорости автотранспортного средства, необходимых действиях водителя для соблюдения ПДД при осуществлении проезда через переезд. Полученная информация будет отображаться на экране дисплея навигатора.**

Ключевые слова: **модель, железнодорожный переезд, шлагбаум, светофор, GPS, ГЛОНАСС, АТС, ДТП, навигатор, навигационная система.**

Железнодорожные переезды являются наиболее сложными и опасными элементами улично-дорожной сети, так как находятся на пересечении железнодорожных путей и автомобильных дорог в одном уровне. В целях обеспечения безопасности дорожного движения железнодорожные переезды оборудуются необходимыми устройствами, обеспечивающими безопасность движения. Являясь объектами повышенной опасности, железнодорожные переезды требуют от участников дорожного движения и работников железных дорог строгого выполнения Правил дорожного движения (ПДД), Правил технической эксплуатации железных дорог РФ, Правил пользования автомобильными дорогами РФ и других нормативных правовых документов. Опасность заключается в том, что автотранспорт и железнодорожный транспорт обладают различными весогабаритными характеристиками, скоростью движения, тормозным путем, приоритетностью проезда через переезд.

Правовые основы обеспечения безопасности дорожного движения на территории РФ определены в федеральном законе от 10.12.1995 № 196-ФЗ (ред. от 26.07.2017) «О безопасности дорожного движения».

Действующие в РФ железнодорожные переезды по месту расположения подразделяются на железнодорожные переезды общего пользования, стоящие на пересечениях железнодорожных путей с автомобильными дорогами общего пользования, муниципальными автомобильными дорогами и улицами, и на железнодорожные переезды необщего пользования, стоящие на пересечениях железнодорожных путей с автомобильными дорогами отдельных предприятий или организаций (независимо от форм собственности). Порядок содержания и обслуживания переездов общего и необщего пользования устанавливается начальником железной дороги. Устройство, оборудование, содержание и обслуживание переездов необщего пользования выполняются за счет средств предприятий, организаций или органов управления автомобильными дорогами и организаций, содержащих автомобильные дороги, пользующихся этими переездами.

Переезды делятся на регулируемые и нерегулируемые [1]. К регулируемым относятся переезды, оборудованные устройствами переездной сигнализации, извещающей водителей транспортных средств о подходе к переезду поезда (подвижного состава), или обслуживаемые дежурными работниками, а также другими работниками железной дороги, которым поручено осуществлять регулирование движения поездов (подвижного состава) и транспортных средств на переезде. К нерегулируемым относятся переезды, не оборудованные устройствами переездной сигнализации и не обслуживаемые дежурными по переезду и другими работниками, которым поручено осуществлять регулирование движения поездов (подвижного состава) и транспортных средств на переезде.

Возможность безопасного проезда через такие переезды определяется водителем транспортного средства в соответствии с Правилами дорожного движения Российской Федерации.

Оборудование действующих переездов устройствами переездной сигнализации осуществляется железными дорогами в соответствии с годовыми и перспективными планами.

Для обеспечения безопасности на железнодорожных переездах России чаще всего используют [2–4]:

- УЗП — устройства заграждения железнодорожного переезда, преграждающие движение автотранспорта через железнодорожный переезд путем подъема специальных плит на проезжей части автомобильной дороги;
- панели дорожного покрытия. Используются в местах пересечения железных и автомобильных дорог в одном уровне. В конструкции применяются резиновые, а также бетонные панели;
- шлагбаумы — устройства для перекрытия проезжей части автомобильной дороги и прекращения движения транспортных средств (участников дорожного движения) через железнодорожный переезд.

Для обеспечения безопасности на железнодорожных переездах за рубежом чаще всего используют:

- ПАКУ — переездное автоматическое контрольное устройство, в котором момент включения светофорной сигнализации на переезде зависит от реальной скорости движения поезда;
- предикторы — высокотехнологичные устройства на базе рельсовых цепей, приводящие в действие переездную автоматику независимо от действующей на линии системы сигнализации. Предикторы используют рельсы для передачи сигналов тональной частоты в обоих направлениях от переезда. Важно, что предикторы могут оценивать скорость движения поездов;
- системы с радиосигналами. Система использует закрепленные на рельсах детекторы, которые посылают радио- или звуковой сигнал наблюдателю. Для подачи сигнала о приближении поезда в случае работы близ шумных путевых машин можно также использовать вибрационные устройства, прикрепляемые к одежде.

Безопасность проезда железнодорожных переездов регламентирована статьей 15 ПДД, в которой говорится, что при подъезде к переезду водитель обязан руководствоваться требованиями дорожных знаков, светофоров, разметки, положением шлагбаума и указаниями дежурного по переезду и убедиться в отсутствии приближающегося поезда (локомотива, дрезины).

Несмотря на огромное количество мероприятий [5], проводимых руководителями железных дорог, МВД, ГИБДД, местными органами власти, направленных на снижение дорожно-транспортных происшествий, количество их не уменьшается, а увеличивается.

Так, по данным Всемирной организации здравоохранения ежегодно в дорожно-транспортных происшествиях погибает около 1,25 млн человек и до 50 млн получают травмы различной степени тяжести [6–8]. Этому способствуют высокие темпы автомобилизации, низкая подготовка водителей автотранспортных средств, несоблюдение установленных ПДД, невнимательность, отвлечение (разговоры по мобильным средствам связи). Так, по данным ОАО «РЖД», в 2018 году на территории РФ находится в эксплуатации более 11 000 железнодорожных переездов, причем чуть менее трети из них обслуживаются непосредственно сотрудниками РЖД. По статистическим данным ОАО «РЖД», всего произошло 211 ДТП, из них по вине водителей АТС — 155, пострадало 108 человек, погиб 31 человек. Основной причиной столкновения между двумя различными транспортными средствами является желание пересечь железнодорожные пути в кратчайшие сроки, игнорируя указания светофоров, закрытых шлагбаумов на переездах, дежурного по переезду.

К проблемным нарушителям ПДД добавилась большая группа водителей-мигрантов с низкой профессиональной подготовкой, незнанием особенностей управления автотранспортными средствами в условиях крупных городов и правил проезда железнодорожных переездов.

С целью обеспечения безопасности дорожного движения, снижения травматизма и недопущения нарушений водителями правил проезда железнодорожных переездов предлагается использовать навигационную систему, функциональную основу которой составляет навигационная программа [9–13], построенная на электронной карте.

Векторные электронные карты поддерживают маршрутизацию, включают множество объектов с их географическими координатами. В пользу применения навигаторов говорят реализованные в них следующие навигационные функции: определение текущего местоположения автотранспортного средства и его отображение на электронной карте, планирование, прокладка и изменение маршрута, отслеживание правильности следования по маршруту и информирование водителя в случае отклонения от маршрута, информационное сопровождение водителя по маршруту и т. д.

Современные навигаторы представляют собой техническое устройство, на экране которого изображена карта местности с нанесенными на нее необходимыми дорожно-инфраструктурными обозначениями и знаками, регулирующими правила подъезда к переезду и порядок его проезда.

В данной статье в качестве дополнительной меры обеспечения безопасности дорожного движения предлагается осуществлять своевременное информирование и предупреждение водителя автотранспортного средства и машиниста поезда о приближении к железнодорожным переездам, об установленных знаках, расстоянии до этих знаков и до переезда, использовать данные навигаторов. Информирование будет осуществляться с помощью встроенных в навигационную систему таких функций, как звуковая сигнализация и голосовое сопровождение.

На рисунке 1 показан алгоритм информационного взаимодействия водителя АТС и навигационной системы на регулируемом железнодорожном переезде, на рисунке 2 — информация, отображаемая на экране дисплея навигатора при проезде через регулируемый железнодорожный переезд. Мы видим на экране монитора, что АТС приближается к нерегулируемому переезду, о чем идет голосовое и звуковое сообщение, показывается требуемая и реальная скорость движения, предложение о необходимости снижения скорости. На правом рисунке показано звуковое и голосовое сообщение о том, что скорость движения сокращается, но сохраняется вероятность возникновения ДТП.

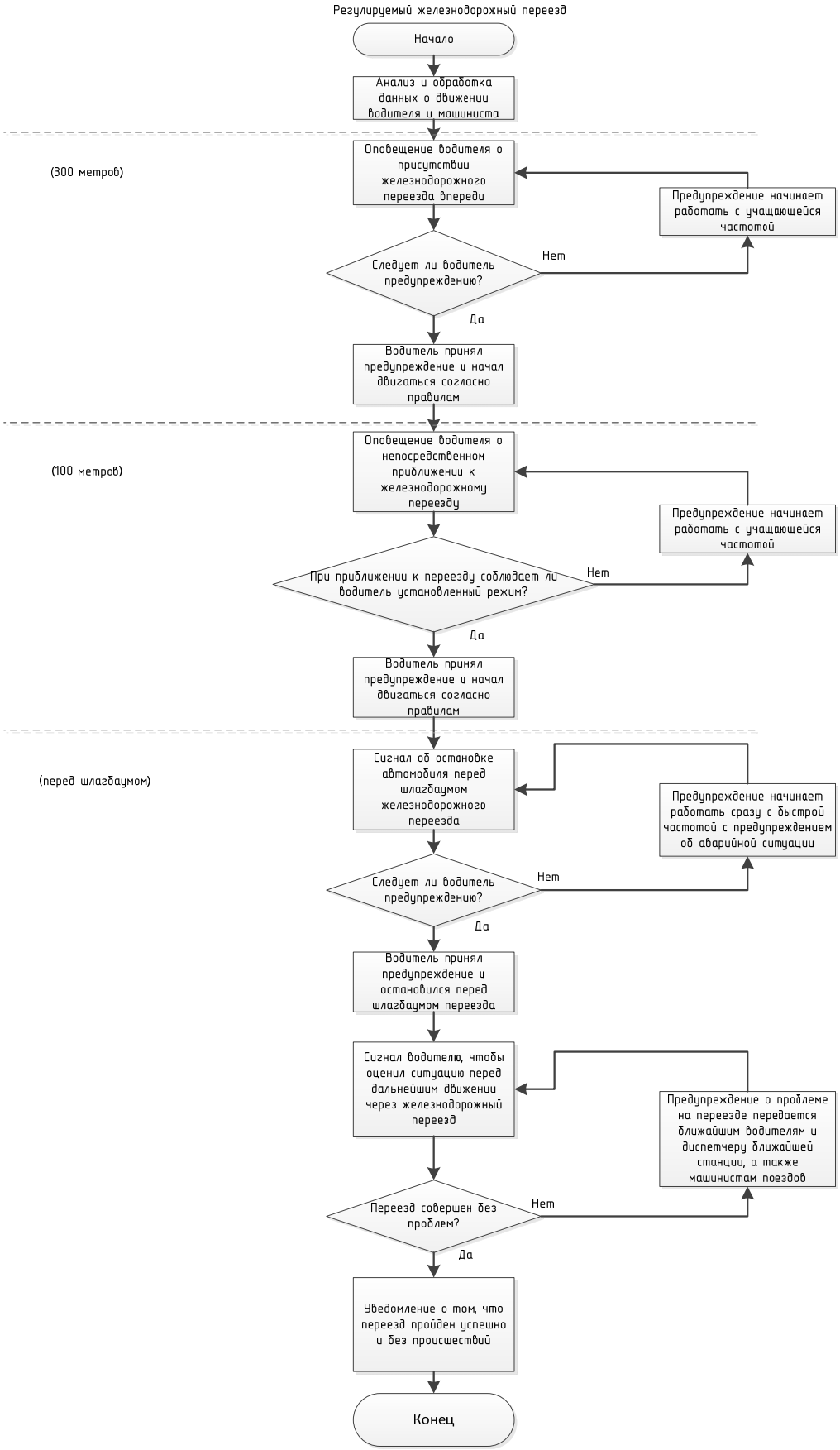


Рис.1. Алгоритм информационного взаимодействия водителя АТС и навигационной системы при проезде через регулируемый железнодорожный переезд

Рис. 2. Информация, отображаемая на экране дисплея навигатора при проезде через нерегулируемый железнодорожный переезд

На рисунке 3 показан алгоритм информационного взаимодействия водителя АТС и навигационной системы при проезде через нерегулируемый железнодорожный переезд. Отображение информации на экране монитора будет соответствовать обстановке, складывающейся возле переезда.

Заключение

С помощью предложенных алгоритмов повысится эффективность безопасного проезда железнодорожных переездов за счет привлечения внимания водителей АТС путем реализации функций звукового и голосового сообщений, отображаемых на экране монитора навигатора.

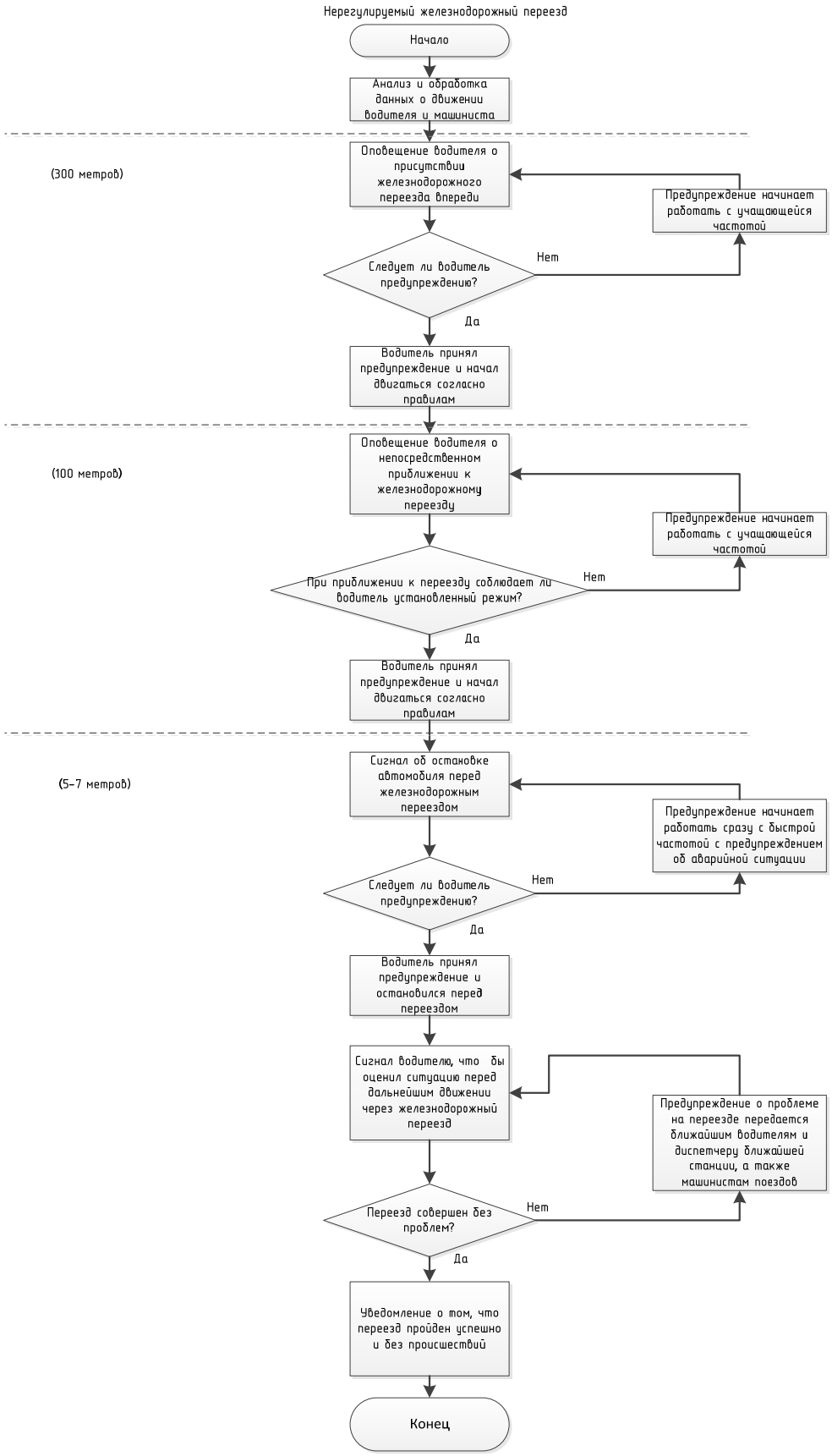


Рис. 3. Алгоритм информационного взаимодействия водителя АТС и навигационной системы при проезде через нерегулируемый железнодорожный переезд

Литература

1. Об утверждении Условий эксплуатации железнодорожных переездов: приказ Министерства транспорта Российской Федерации от 31.07.2015 № 237.

2. О федеральной целевой программе «Повышение безопасности дорожного движения в 2013–2020 годах»: постановление Правительства Российской Федерации от 03.10.2013 № 864 (ред. от 13.12.2017).

3. Шаповал О. Л. Обеспечение безопасности на железнодорожных переездах // Системы безопасности. – 2011. — № 2. — С. 118–119.

4. Рожанский Д.В. Повышение безопасности движения в зоне железнодорожных переездов / Д.В. Рожанский, С.Н. Карасевич // Вестник Белорусского национального технического университета. — 2007. — № 2. — С. 60–65.

5. Миненко Е.Ю. Анализ мероприятий, направленных на решение проблемы безопасности на железнодорожных переездах / Е.Ю. Миненко, Ю.А. Кусморова // Молодой учёный. — 2014. — № 17 (76). — С. 78–80.

6. Переезды Российских железных дорог: аналитические материалы о переездах железных дорог и обеспечении безопасности движения на них / Департамент пути и сооружений ОАО «РЖД». — М. : Академкнига, 2004. — 152 с.

7. Карпущенко Н.И. Проблема обеспечения безопасности на железнодорожных переездах / Н.И. Карпущенко, Д.В. Величко, Т.В. Колмогорова // Транспорт Российской Федерации. — 2011. — № 4 (35). — С.47–50.

8. Тройнікова О.М. Підходи до управління безпекою траспортних процесів на залізниці. // Энергосбережение. Энергетика. Энергоаудит. — 2015. — № 1(132). — С. 54–60.

9. Коваленко В.Н. Современные тенденции автоматизации переездов на железнодорожном транспорте / В. Н. Коваленко, М. Н. Катаев // Инновационный транспорт. 2015. — № 3 (17). — С. 54–58.

10. Федухин А.В. Информационный подход к повышению безопасности движения на железнодорожных переездах / А.В. Федухин, Ар.А. Муха // Математічни машини і системи. — 2015. — № 4. — С. 145–151.

11. Tomis, M., Dvorsky, M., Styskala V., et al. Wireless Barrage on the Railway Crossing, *38th International Conference on Telecommunications and Signal Processing (TCP)* (9–11 July 2015, Prague, Czech Republic), pp. 129–133.

12. Об оснащении транспортных, технических средств и систем аппаратурой спутниковой навигации ГЛОНАСС или ГЛОНАСС/GPS: постановление Правительства Российской Федерации от 25.08.2008 № 641 (с изм. и доп. от 17.12.2010, 12.11.2016).

13. Об использовании в Российской Федерации глобальных навигационных спутниковых систем на транспорте : приказ Министерства транспорта Российской Федерации от 03.09.2012 № 63.

Safety at Level Crossings by Using GPS/GLONASS

Student T.I. Vaskov, Student E.A. Mihaylenko, PhD R.G. Gilvanov
Emperor Alexander I St. Petersburg State Transport University
St. Petersburg, Russia
timoff.pk@yandex.ru, mihaylenko.evgeniy77@gmail.com, rinatgilvanov@mail.ru

Abstract. **The main purpose of safety at level crossings is decrease the number of accidents by adding to the navigation systems auxilliary signs (objects) which represent regulated and non-regulated level crossings. In this case drivers will get additional sound and voice information about approach to the level crossing, type and state of it, allowed and current speed of the car, necessary actions while passage of the level crossing. This information will be shown on the display of the navigation device.**

Keywords: **model, level crossing, gate, traffic lights, GPS, GLONASS, accidents, navigation, navigation systems.**

REFERENCES

1. About the Statement of Operating Conditions of Railway Crossings: Order of the Ministry of transport of the Russian Federation [Ob utverzhdenii usloviy ekspluatatsii zheleznodorozhnykh pereezdov: Prikaz Ministerstva transporta Rossiyskoy Federatsii], from 31.07.2015 No. 237.

2. About the Federal Target Program «Increase of Traffic Safety in 2013–2020»: Resolution of the Government of the Russian Federation [O federal`noy tselevoy programme «Povyshenie bezopasnosti dorozhnogo dvizheniya v 2013–2020 godakh»: Postanovlenie Pravitel'stva Rossiyskoy Federatsii] from 03.10.2013 No. 864 (ed. by 13.12.2017).

3. Shapoval O.L. Ensuring safety at railway crossings [Obespechenie bezopasnosti na zheleznodorozhnykh pereezdakh], *Security and Safety [Sistemy bezopasnosti]*, 2011, No. 2, pp. 118–119.

4. Rozhansky D.V., Karasevich S.N. Improvement of Traffic Safety in the Zone of Railway Crossing [Povyshenie bezopasnosti dvizheniya v zone zheleznodorozhnykh pereezdov], *Bulletin of the Belarusian National Technical University [Vestnik Belorusskogo Natsional`nogo Tekhnicheskogo Universiteta]*, 2007, No. 2, pp. 60–65.

5. Minenko E.Yu., Kusmorova Yu.A. Analysis of Measures Aimed at Solving the Problem of Safety at Railway Crossings [Analiz meropriyatiy, napravlennykh na reshenie problemy bezopasnosti na zheleznodorozhnykh pereezdakh], *Young Scientist [Molodoy uchenyy]*, 2014, No. 17 (76), pp. 78–80.

6. The crossings of the Russian Railways: analysis at railway crossings and traffic safety [Pereezdy Rossijskikh zheleznykh dorog: analiticheskie materialy o pereezdakh zheleznykh dorog i obespechenii bezopasnosti dvizheniya na nikh], Moscow, Academic Book Publishing House, 2004, 152 p.

7. Karpushchenko N.I., Velichko D.V., Kolmogorova T.V. Problem of Ensuring Traffic Safety in Railway Crossings [Problema obespecheniya bezopasnosti na zheleznodorozhnykh pereezdakh], *Transport of the Russian Federation* [Transport Rossiyskoy Federatsii], 2011, No. 4 (35), pp. 47–50.

8. Troynikova H.N. Approaches to Safety Management of Transport Processes on the Railway [Podkhody k upravleniyu bezopasnost'yu transportnykh protsessov na zheleznoy doroge], *Energy saving. Power engineering. Energy audit [Energosberezhenie. Energetika. Energoaudit]*, 2015, No. 1 (132), pp. 54–60.

9. Kovalenko V. N., Kataev M. N. Modern Trends in Automation of Level Crossings on Railway Transport [Sovremennye tendencii avtomatizacii pereezdov na zheleznodorozhnom transporte], *Innotrans [Innovacionnyj transport]*, 2015, No. 3 (17), pp. 54–58.

10. Fedukhin O.V., Mukha Ar.A. Information approach to Improving Traffic Safety at Railway Crossings [Informatsionnyy podkhod k povysheniyu bezopasnosti dvizheniya na zheleznodorozhnykh pereezdakh] // *Mathematical Machines and Systems [Matematicheskie mashiny i sistemy]*, 2015, No. 4, pp. 145–151.

11. Tomis, M., Dvorsky, M., Styskala V., et al. Wireless Barrage on the Railway Crossing, *38th International Conference on Telecommunications and Signal Processing (TCP)* (9–11 July 2015, Prague, Czech Republic), pp. 129–133.

12. Equipment of Transport and Technical Means and Systems with Satellite Navigation GLONASS/GPS: Resolution of the Government of the Russian Federation [Ob osnashchenii transportnykh, tekhnicheskikh sredstv i sistem apparaturoy sputnikovoy navigatsii GLONASS ili GLONASS/GPS: Postanovlenie Pravitel`stva Rossiyskoy Federatsii] of 25.08.2008 No. 641 (as amended on 17.12.2010 and 12.11.2016).

13. About Use in the Russian Federation of Global Navigation Satellite Systems on Transport: Order of the Ministry of transport of the Russian Federation [Ob ispol'zovanii v Rossiyskoy Federatsii global'nykh navigacionnykh sputnikovykh sistem na transporte: Prikaz Ministerstva transporta Rossiyskoy Federatsii] of 03.09.2012, No. 63.

Синтаксический анализ кода HLASM в среде разработки Intellij IDEA

студент магистратуры К.А. Хасанов, студент магистратуры В.И. Каракозов
Петербургский государственный университет путей сообщения Императора Александра I
Санкт-Петербург, Россия
hasanow.kamil@yandex.ru

Аннотация. **Рассматриваются решения по синтаксическому разбору кода на языке HLASM с последующей подсветкой синтаксиса, применяемой в среде разработки Intellij IDEA для повышения эффективности работы программиста. При анализе этой части кода можно выделить общий модуль, в котором объявлены инструкции, в которых имеются некоторые параметры. Модуль нужно отразить в виде формализованной модели, которая показывает типы, значения, вложенность кода и сами сущности. Для этого используется синтаксическое дерево. Семантическая модель кода и дерево разбора строятся с помощью лексического анализатора и парсера, которые понимают грамматику выбранного языка программирования, проходя от вершины к конечным операндам. Описывается характеристика программной реализации. Приводится схема иерархии правил. Предложенное решение позволяет повысить удобство процесса написания кода.**

Ключевые слова: **код, HLASM, синтаксический анализ, Intellij IDEA, триплет, грамматика, синтаксическое дерево, лексер, парсер, Java, ANTLR.**

Введение

В современном мире многие компании занимаются разработкой программного кода. Для того чтобы упростить и поддерживать процесс создания и отладки кода, разрабатываются среды программирования, которые являются востребованными на рынке программного обеспечения. Большое количество популярных и лидирующих языков программирования распространяются в виде интегрированных сред, порой являющихся многофункциональными, которые упрощают и предоставляют программисту доступ к возможностям языка. Сложность современных интерпретаторов, компиляторов и архитектур требует от сред разработки все большей автоматизации, исключения из работы программиста рутинной технической работы.

Интегрированные среды разработки дают большие преимущества для программистов, которые только начали знакомство с языком, а также повышают эффективность написания кода [1]. Любая крупная среда разработки имеет вспомогательные средства, такие как выделение структуры исходного кода, контекстно-зависимые подсказки и помощь, расцветка синтаксиса.

К сожалению, процесс разработки и внедрения поддержки новых языков программирования с использованием практически любой из существующих сред является сложной и трудоемкой задачей.

В данной работе представлена работа с Intellij IDEA SDK — инструментарием для разработки приложения, в частности, и редакторов кода. Intellij IDEA — это мультиязычная интегрированная среда разработки программного обеспечения, таких как Python, Java, JavaScript, созданная компанией JetBrains.

На основе синтаксического анализа текста программы выполняется расцветка текста, которая предоставляется библиотекой Intellij IDEA SDK. Процесс расцветки — это выделение значимых лексем и сопоставление им визуальных атрибутов, таких как размер, шрифт, цвет. Помимо визуальных атрибутов предоставляется анализ структуры текста: рекурсивных и парных конструкций; выделение структурного дерева позволяет целевой IDE реализовать расширенную навигацию по тексту.

Язык ассемблера высокого уровня HLASM

В качестве языка, для кода которого проводится синтаксический анализ, выбран язык ассемблера высокого уровня (High-Level Assembler, HLASM). Он представляет собой язык ассемблера высокого уровня фирмы IBM, используемый под управлением операционных систем z/Architecture на компьютерах мейнфрейм. Компьютер такого типа представляет собой большой отказоустойчивый высокопроизводительный универсальный сервер со значительными ресурсами ввода-вывода, большим объемом внешней и оперативной памяти, предназначенный для использования в критически важных системах с интенсивной пакетной и оперативной транзакционной обработкой.

Основанный на ассемблере IBM H, он позволяет программистам писать на ассемблере код, который использует ряд особенностей, связанных с языками высокого уровня. Например, наличие директив, зависимых и отмеченных от USING, более полная перекрестно-ссылочная информация, а также дополнительные макроязыковые средства (например, для разработки пользовательских функций) [2].

В операционных системах z/Architecture имеется редактор ISPF (Interactive System Productivity Facility), внешний вид которого представлен на рисунке 1. Этот редактор имеет лишь базовую подсветку синтаксиса.

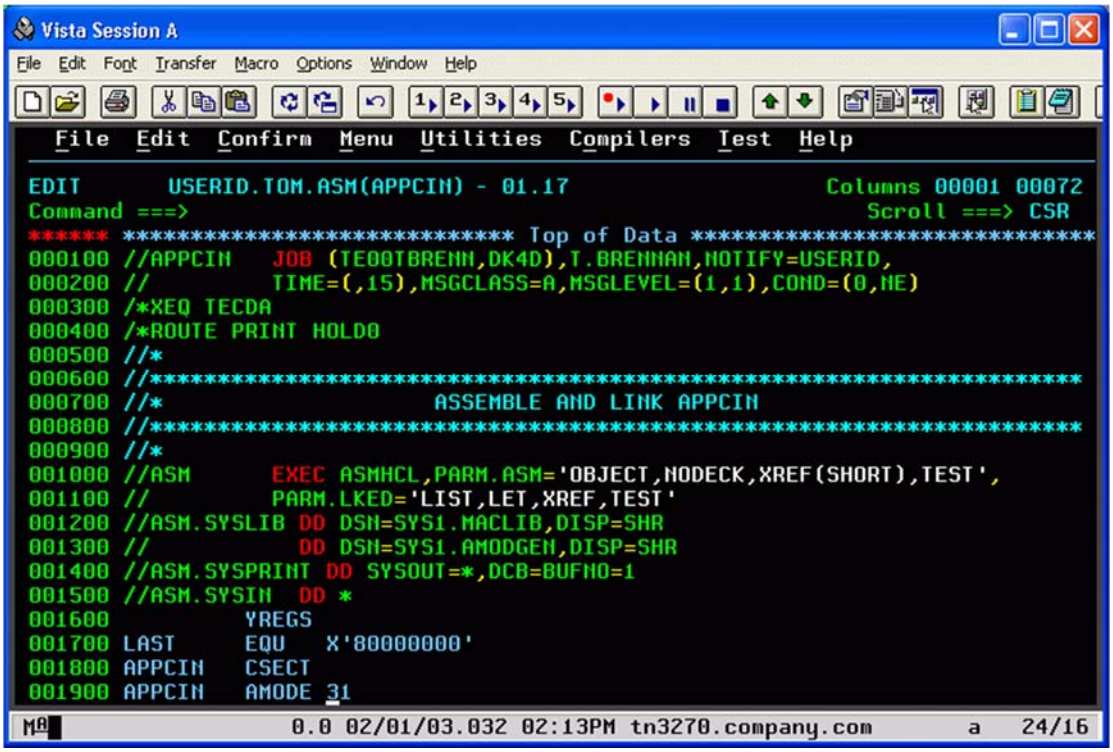


Рис. 1. Редактор ISPF

В настоящее время не существует сред разработки с поддержкой языка HLASM. Поэтому создание программного продукта, который будет разбирать код HLASM и подсвечивать его, а также дальнейшее расширение функционала с помощью средств Intellij IDEA SDK позволит облегчить процесс написания кода для программистов.

Синтаксический анализ программного кода

Выбрав язык программирования, можно перейти к синтаксическому анализу программного кода.

У любого языка программирования имеется свой алфавит, состоящий из нетерминальных и терминальных символов. Нетерминальным символом является элемент конструкции языка, который не имеет заранее известного значения, такой как формула или команда. Терминальным символом является любой имеющий конкретное известное значение символ, например цифра или буква. Множество терминальных символов образуют нетерминальные.

Любой язык программирования можно описать набором правил, которые выделяют некоторое подмножество известных символов из множества слов конечного алфавита языка. Данный набор правил называется грамматикой языка, которая задает правила, определяющие правильность построения слова языка, и позволяет построить любое новое слово языка. Первые являются распознающими, или аналитическими, грамматиками, вторые являются порождающими [3].

Код, написанный на HLASM, будучи правильно оформленным, представляет собой последовательность инструкций. Соблюдая правила именования сущностей в коде и его правильной структуры, можно получить объектную модель, которая отражает организацию кода. В программе каждый объект можно рассматривать атомарно, не вдаваясь в подробности его реализации — с использованием абстрагирования. Также любой объект может являться подмножеством других объектов, которые можно рассматривать как конечную атомарную сущность. Элементарные типы, включающиеся в объект помимо других, всегда являются конечными и не могут инкапсулировать в себе что-либо. Каждый объект имеет связь с другими, например, имеет некую вложенность в другой объект.

Рассмотрим следующий пример кода:

```
...
LA      R2,3
LA      R3,5
AR      R2,R3
...
```

Эта часть кода записывает значение 3 и 5 в регистры R2 и R3 соответственно. Затем складывает значение этих регистров, сохраняя получившееся в R2.

При анализе этой части кода можно выделить общий модуль, в котором объявлены инструкции, в которых имеются некоторые параметры. Отсюда следует, что это можно и нужно отразить не словами, а в виде некой формализованной модели, которая покажет типы, значения, вложенность кода и сами сущности. Для этого лучшим образом подойдет синтаксическое дерево, каждый узел которого может быть конечным, а именно может отражать поле любого примитивного типа или некое детерминированное значение, либо расходиться вниз, показывая, что элемент не является конечным в иерархии. Пример дерева, построенного для приведенного выше кода, представлен на рисунке 2.

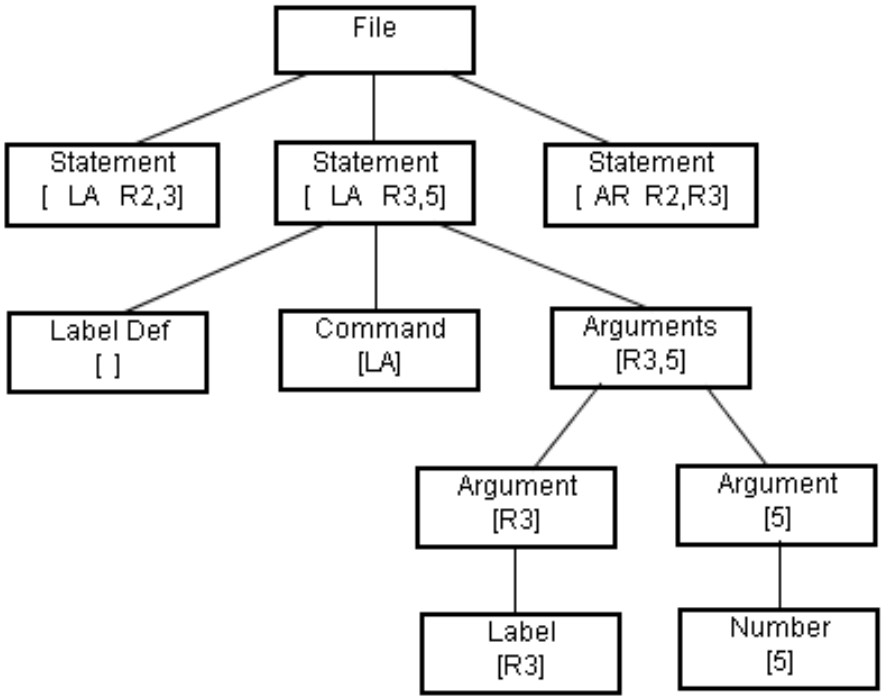


Рис. 2. Дерево разбора кода

Таким образом, при синтаксическом анализе кода можно получить иерархию и элементарных типов, и объектов, а также провести связи и зависимости между ними. Такую иерархию представляет абстрактное синтаксическое дерево (АСД), или Abstract Syntax Tree (AST), ветви которого представляют объекты в коде, а листья — элементарные типы, представленные в коде. Абстрактное синтаксическое дерево — это конечное, помеченное, ориентированное дерево, листья которого сопоставлены с соответствующими операндами, а вершины — с операторами языка программирования. Отсюда следует, что листья представляют только константы и переменные, а также являются лишь пустыми операторами. Количество ветвей, как и количество листьев, может быть любым, однако дерево всегда состоит из одной вершины — объекта, который представляет программу целиком.

В абстрактном синтаксическом дереве элементы могут не определяться конкретной грамматикой разбираемого языка. Классическим примером являются ограничительные скобки в языковых конструкциях — группировка операндов в АСД явно задается структурой дерева, а ограничивающие скобки вообще отсутствуют в разборе, так как не влияют на АСД. Большое количество правил грамматики создают вершину, а символы в правиле становятся ребрами. Правила могут ничего не привносить в АСД. Примером могут служить группирующие, которые просто заменяются в вершине одним из своих символов. Также анализатор может создать полное дерево разбора, а после пройти по нему, попутно удаляя узлы и ребра, не использующиеся в абстрактном синтаксисе, чтобы получить АСД [4].

Решение задачи синтаксического анализа

Построить семантическую модель кода и дерево разбора должны лексический анализатор и парсер, которые понимают грамматику выбранного языка программирования, проходя от вершины к конечным операндам. Лексический анализатор определяет, как содержимое файла будет разбито на последовательность токенов, т. е. нетерминальных символов языка программирования. К примеру, токенами языка Java для полной грамматики языка будут являться блок кода, блок комментариев, метод, класс. Лексический анализатор служит фундаментом почти для всех функций языковых сред, таких как подсветка синтаксиса, функции анализа кода и другие. API лексического анализатора для Intellij IDEA SDK определен в интерфейсе Lexer, а для парсера — в интерфейсе Parser.

IDEA вызывает лексический анализатор в трех основных контекстах:

- подсветка синтаксиса;
- построение абстрактного синтаксического дерева;
- построение индекса слов, которые содержатся в файле, — если используется реализация сканера, основанная на пользовательском лексическом анализаторе.

Лексический анализатор для языка HLASM должен выделять в коде следующие сущности, описывающиеся лексическими правилами или грамматикой:

- директивы;
- инструкции;
- макросы;
- метки;
- комментарии;
- числа;
- простые символы;

– различные атрибуты любой из перечисленных сущностей.

Парсер — это программный модуль, основывающийся на поступающих из потока лексического анализатора токенах, который по заданным правилам строит абстрактное дерево разбора. Ввиду того, что исходный код языка HLASM состоит из последовательности инструкций, директив и макрокоманд, весь исходный код можно разбить на отдельные блоки, состоящие из полного описания макроса или отдельных инструкций и директив. Они, свою очередь, состоят из метки, названия команды и аргументов. Аргументы разделяются запятыми и состоят из токенов или комбинации токенов.

Программная реализация

Существует множество программных библиотек, которые помогают разобрать текст программы и получить его абстрактное представление в виде дерева. Каждое такое средство разбора оперирует грамматикой выбранного языка, проводя лексический и синтаксический разбор.

Одним из подобных программных средств является Another Tool for Language Recognition (ANTLR). ANTLR — это генератор лексических анализаторов и парсеров на языке Java, принимающий на вход файл с исходным кодом разбираемой программы, а на выходе создающий абстрактное синтаксическое дерево.

Определимся с набором правил. Так как исходный код языка HLASM состоит из последовательности инструкций, то обозначим корневой элемент, или линии (lines), который состоит из одного или более заявлений (statement). Каждое заявление, в свою очередь, состоит из макроса (macro) или целой линии (line_wrapper).

Целая линия раскрывается в последовательность, которая состоит из метки по умолчанию (label_def), линии (line) и окончания целой линии (endline). Причем линии и метки по умолчанию может и не быть. Метка по умолчанию является конечным элементом.

Линия состоит из команды (commands) и директивы (directive), которые являются конечными элементами, а также аргументов (arguments). Аргументы разделены между собой конечным символом ','.

Аргумент может быть *именованным аргументом* (named_argument) или некоторым значением (expression). Значение является токеном или комбинацией токенов. Именованный аргумент состоит из последовательности метки (label), знака '=' и значения.

Макрос состоит из последовательности: 'MACRO', аргументы, окончание строки, описание макроса (macro_def_wr), линии, метка по умолчанию, 'MEND', аргументы, окончание строки.

Описание макроса включает в себя целую линию.

Примерная схема иерархии правил представлена на рисунке 3.

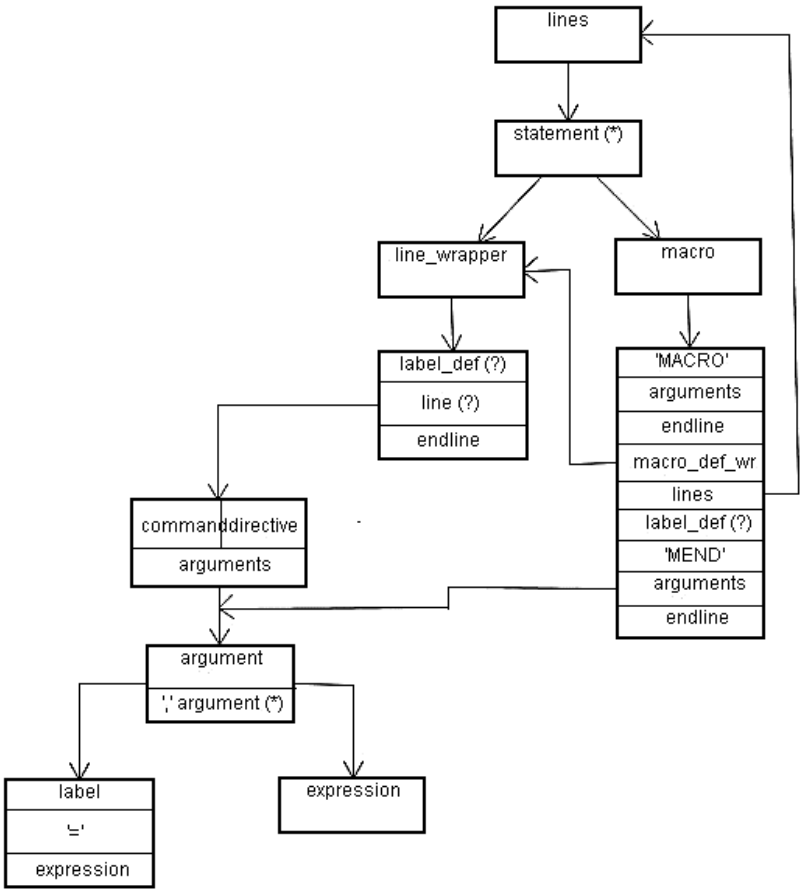


Рис. 3. Иерархия правил

Следующим этапом является подсветка синтаксиса и ошибок, которая выполняется на разных уровнях. На первом — подсветка синтаксиса, которая основана на результатах лексического разбора, осуществляется посредством интерфейса SyntaxHighlighter. Этот интерфейс возвращает экземпляры TextAttributeKey для каждого типа токенов, который требует особую подсветку. Для подсвечивания ошибок лексического анализатора применяется стандартный объект класса TextAttributeKey для недопустимых символов (HighlighterColors.BAD_CHARACTER). На втором уровне подсветки — выделение ошибок, произошедших во время синтаксического разбора с определением цепочки токенов, которые не соответствуют грамматике языка [5].

Пример подсветки синтаксиса в среде разработке Intellij IDEA представлен на рисунке 4.

Весь процесс разбора кода и его подсветки, как «черный ящик», должен получать на входе исходный код программы, а на выходе — подсвеченный код, при необходимости — подсвеченные ошибки, которые не соответствуют синтаксису языка.

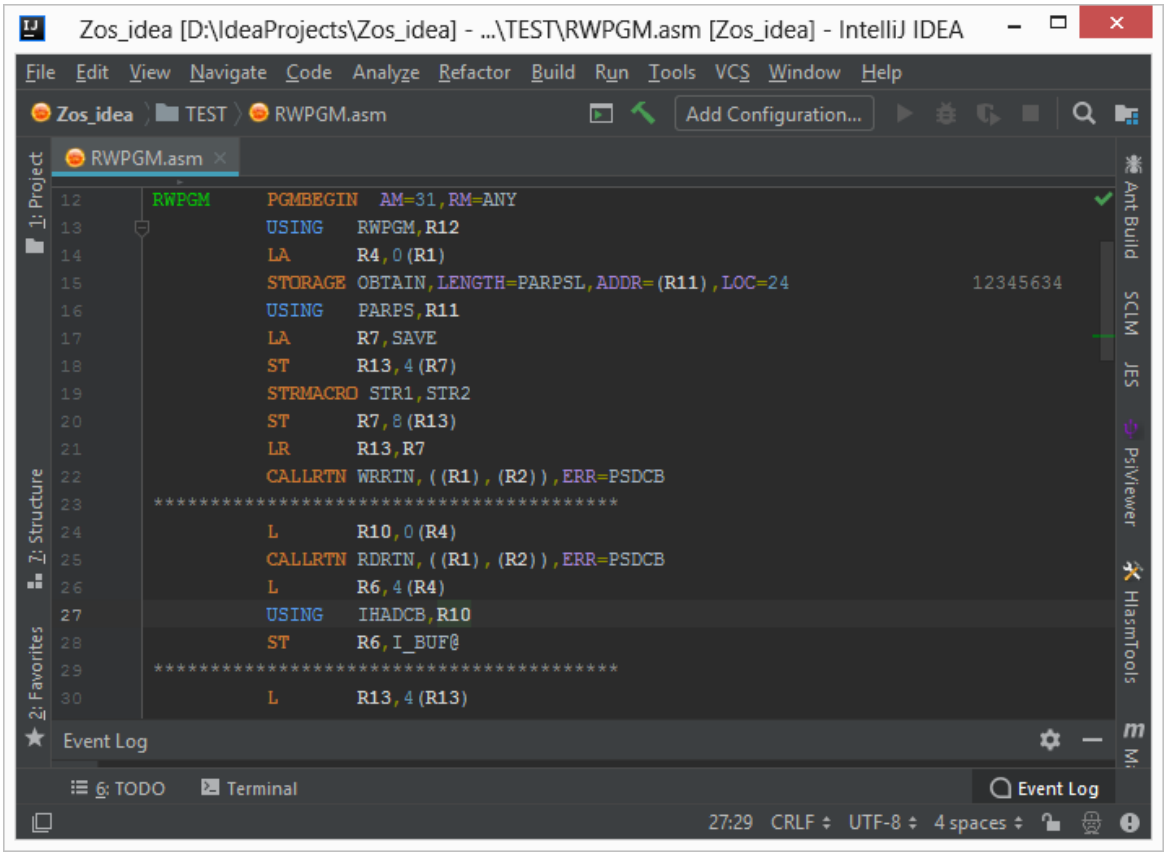


Рис. 4. Пример подсветки кода HLASM в Intellij IDEA

Заключение

В работе рассмотрены основные принципы решения задачи синтаксического анализа кода HLASM и его базовая подсветка синтаксиса, которые были реализованы в среде разработке Intellij IDEA. Эта среда разработки поддерживает большое количество функционала, который можно добавить в этот программный продукт. Предложенное решение может найти достаточно широкое применение при разработке программ на языке HLASM, позволяя повысить удобство процесса написания кода.

Литература

1. Thomas R.C. Long Term Human-Computer Interaction: An Exploratory Perspective. London, Springer, 1998, 212 p.

2. HLASM // Википедия. [2007–2019]. URL: http://ru.wikipedia.org/wiki/HLASM (дата обращения: 11.03.2019).

3. Context-free grammar // Wikipedia [2001–2019]. URL: http://en.wikipedia.org/wiki/Context-free_grammar (дата обращения: 11.03.2019).

4. Parr T. The Definitive ANTLR Reference: Building Domain-Specific Languages. Raleigh, Dallas, The Pragmatic Bookshelf, 2007, 384 p.

5. Подсветка синтаксиса и ошибок / Разработка плагина IntelliJ IDEA. Ч. 5 // Habr. URL: http://habr.com/ru/post/187292 (дата обращения: 11.03.2019).

Syntax Parsing for HLASM Language in the Development Environment Intellij IDEA

Graduate Student K.A. Khasanov, Graduate Student V.I. Karakozov
Emperor Alexander I St. Petersburg State Transport University
Saint Petersburg, Russia
hasanow.kamil@yandex.ru

Abstract. **Considering solutions for parsing code in the HLASM programming language with syntax highlighting, that used in the development environment Intellij IDEA to improve the efficiency of the programmer. When analyzing this part of code, the main we can identify a main module that declares statements that have some parameters. The module needs to be reflected in the form of a formalized model which shows the types, values, nesting of the code and the entities. The syntax tree is used for this purpose. The semantic code model and parsing tree are built with the help of a lexical analyzer and parser that understand the grammar of the selected programming language, passing from the top to the final operands. The characteristic of software implementation is described. The scheme of the rule hierarchy is given. The proposed solution improves the convenience of the code writing process.**

Keywords: **code, HLASM, syntax analys, Intellij IDEA, triplet, grammar, syntax tree, parser, Java, ANTLR.**

REFERENCES

1. Thomas R.C. Long Term Human-Computer Interaction: An Exploratory Perspective. London, Springer, 1998, 212 p.
2. HLASM, *Wikipedia*. Available at: http://ru.wikipedia.org/wiki/HLASM (accessed 11 March 2019).
3. Context-free grammar, *Wikipedia*. Available at: http://en.wikipedia.org/wiki/Context-free_grammar (accessed 11 March 2019).
4. Parr T. The Definitive ANTLR Reference: Building Domain-Specific Languages. Raleigh, Dallas, The Pragmatic Bookshelf, 2007, 384 p.
5. Syntax and error highlighting [Podsvetka sintaksisa i oshibok]. *In: IntelliJ IDEA plugin development. Part 5 [Razrabotka plagina IntelliJ IDEA. Chast` 5]*, *Habr.* Available at: http://habr.com/ru/post/187292 (accessed 11 March 2019).