

ISSN 1311-3321 (print)
ISSN 2535-1028 (CD-ROM)
ISSN 2603-4123 (on-line)

UNIVERSITY OF RUSE “Angel Kanchev”
РУСЕНСКИ УНИВЕРСИТЕТ “Ангел Кънчев”

BSc, MSc and PhD Students & Young Scientists
Студенти, докторанти и млади учени

PROCEEDINGS

Volume 57, book 6.5.
Financial Mathematics
&
Informatics

НАУЧНИ ТРУДОВЕ

Том 57, серия 6.5.
Финансова математика
&
Информатика

Ruse
Русе
2018

Volume 57 of PROCEEDINGS includes the papers presented at the scientific conference RU & SU'18, organized and conducted by University of Ruse "Angel Kanchev" and the Union of Scientists - Ruse. Series 6.5. contains papers reported in the Financial Mathematics & Informatics section.

Book	Code	Faculty and Section
Faculty of Natural Sciences and Education		
6.1	FRI-1.414-1-MIP	Mathematics, Informatics and Physics
6.2	FRI-2G.307-1-PP	Pedagogy and Psychology
	FRI-2.207-1-HEF	History, Ethnology and Folklore
6.3	FRI-2G.405-1-LL	Linguistics and Literature
	FRI-2G.407-1-AS	Art Studies
6.4	FRI-2G.305-1-ERI	Education - Research and Innovations
6.5	WED-2.203-SSS-FM	Financial Mathematics
	THU-1.416-SSS-I	Informatics

The papers have been reviewed.

ISSN 1311-3321 (print)

ISSN 2535-1028 (CD-ROM)

ISSN 2603-4123 (on-line) Copyright © authors

The issue was included in the international ISSN database, available at <https://portal.issn.org/>.

The online edition is registered in the portal ROAD scientific resources online open access



PROGRAMME COMMITTEE

- **Prof. Velizara Pencheva, PhD,**
University of Ruse, Bulgaria
- **Prof. Leon Rothkrantz**
Delft University of Technology, Netherlands
- **Assoc. Prof. Antonio Jose Mendes,**
University of Coimbra, Portugal
- **Prof. Ville Leppanen,**
University of Turku, Finland
- **Assoc. Prof. Marco Porta,**
University of Pavia, Italy
- **Prof. Douglas Harms,**
DePauw University, USA
- **Prof. Ismo Hakala, PhD,**
University of Jyväskylä, Finland
- **Prof. Dr. Artur Jutman,**
Tallinn University of Technology, Estonia
- **Prof. RNDr. Vladimír Tvarozek, PhD,**
Slovak University of Technology in Bratislava, Bratislava, Slovakia
- **Doc. Ing. Zuzana Palkova, PhD,**
Slovak University of Agriculture in Nitra, Nitra, Slovakia
- **Andrzej Tutaj, PhD,**
AGH University of Science and Technology, Krakow, Poland
- **Assoc. Prof. Behiç TEKİN, PhD,**
EGE University, Izmir, Turkey,
- **Prof. Valentin NEDEFF Dr. eng. Dr.h.c.,**
“Vasile Alecsandri” University of Bacău, Romania
- **Dr. Cătălin POPA,**
“Mircea cel Bătrân” Naval Academy, Constantza, Romania
- **Prof. dr Larisa Jovanović,**
Alfa University, Belgrade, Serbia
- **Prof. dr hab. Edmund LORENCOWICZ,**
University of Life Sciences in Lublin, Poland
- **Assoc. Prof. Ion MIERLUS - MAZILU, PhD,**
Technical University of Civil Engineering, Bucharest, Romania
- **Prof. Dojčil Vojvodić PhD,**
Faculty of Philosophy, University of Novi Sad, Serbia
- **Assoc. Prof. Alexandrache Carmen, PhD,**
Department of Teacher Training, “Dunarea de Jos”, Galati University, Romania
- **Prof. Alberto Cabada,**
University of Santiago de Compostela, Faculty of Mathematics, Santiago de Compostela, Spain
- **Assoc. Prof. Dr. Mehmet Şahin,**
Necmettin Erbakan University, Ahmet Keleşoğlu Faculty of Education, Konya, Turkey
- **Assoc. Prof. Erika Gyöngyösi Wiersum, PhD,**
Eszterházy Károly University, Comenius Campus in Sárospatak, Institute of Real Sciences, Sárospatak, Hungary
- **Anna Klimentova, PhD,**
Constantine the Philosopher University in Nitra, Slovakia
- **Prof. Igor Kevorkovich Danilov, DSc,**
Yuri Gagarin State Technical University of Saratov, Russia

- **Prof. Aleksander Valentinov Sladkowski, DSc,**
Silesian University of Technology, Poland
- **Prof. Pether Shulte, PhD,**
Institute for European Affairs (INEA), Dusseldorf, Germany
- **Prof. Aslitdin Nizamov, DSc., PhD,**
Bukhara Engineering-Technological Institute, Bukhara, Uzbekistan
- **Prof. Marina Sheresheva, PhD,**
Lomonosov Moscow State University, Russia
- **Prof. Erik Dahlquist, PhD,**
Mälardalen University, Sweden
- **Prof. Erik Lindhult, PhD,**
Mälardalen University, Sweden
- **Prof. Annika Kunnasvirta, PhD,**
Turku University of Applied Sciences, Finland
- **Prof. Walter Leal, Dr. (mult.) Dr.h.c. (mult.),**
Hamburg University of Applied Sciences, Germany
- **Prof. Dr. Gerhard Fiolka,**
University of Fribourg, Switzerland
- **Prof. Haluk Kabaalioglu, PhD,**
Yeditepe University, Turkey
- **Prof. Silva Alves, PhD,**
University of Lisbon, Portugal
- **Hanneke van Bruggen,**
Appeldoorn, The Netherlands
- **Nino Žganec,**
President of European Association of Schools of Social Work, Assoc. Prof. at the Department of Social Work,
University of Zagreb, Croatia
- **Prof. Violeta Jotova,**
Направление „Педиатрия“ в УМБАЛ „Св. Марина“ – Варна, България
- **Assoc. Prof. Tanya Timeva, MD, PhD,**
Obstetrics and Gynecology Hospital "Dr. Shterev", Sofia, Bulgaria
- **Prof. Kiril Stoychev, PhD,**
Institute of Metal Science, Equipment and Technologies “Acad. A. Balevsci” with Hydroaerodynamics centre
– BAS, Bulgaria
- **Assoc. Prof. Mark Shamtsyan, PhD,**
Technical University, Saint Petersburg, Russia
- **Assoc. Prof. Oleksii Gubenia, PhD,**
National University of Food Technologies, Kiev, Ukraine
- **Assoc. Prof. Olexandr Zaichuk, DSc,**
Ukrainian State University of Chemical Technology, Dnepropetrovsk, Ukraine
- **Prof. Eugene Stefanski, DSc,**
Samara University, Russia
- **Doc. Dr. Tatiana Strokovskaya,**
International University of Nature “Dubna”, Dubna, Russia
- **Prof. DSc. Petar Sotirow,**
Maria Curie-Skłodowska University of Lublin, Poland
- **Prof. Papken Ehasar Hovsepian,**
Sheffield Hallam University, Sheffield, UK
- **Assoc. Prof. Krassimir Dochev Dochev, PhD,**
University of Portsmouth School of Engineering, UK
- **Mariana Yordanova Docheva, PhD,**
University of Portsmouth School of Engineering, UK

- **Assoc. Prof. Ivan Antonov Lukanov, PhD,**
University of Botswana, Faculty of Engineering and Technology, Gaborone, Botswana
- **Assoc. Prof. Petko Vladev Petkov, PhD,**
Research Associate Cardiff University, UK
- **Prof. Stepan Terzian, DSc,**
Bulgarian Academy of Science, Bulgaria
- **Prof. Dr. Gabriel Negreanu,**
University Politehnica of Bucharest, Romania

ORGANISING COMMITTEE

- ◆ **ORGANIZED BY: University of Ruse (UR) and Union of Scientists (US) - Ruse**
- ◆ **ORGANISING COMMITTEE:**
 - **Chairpersons:**
COR. MEM Prof. Hristo Beloev, DTSc – Rector of UR, Chairperson of US - Ruse
 - **Scientific Secretary:**
Prof. Diana Antonova PhD, Vice-Rector Research,
dantonova@uni-ruse.bg, 082/888 249
- ◆ **MEMBERS:**
Assoc. Prof. Kaloyan Stoyanov, PhD
Assoc. Prof. Velina Bozduganova, PhD,
Assoc. Prof. Kiril Sirakov, PhD,
Assoc. Prof. Milko Marinov, PhD,
Pr. Assist. Elena Ivanova, PhD,
Assoc. Prof. Simeon Iliev, PhD,
Assoc. Prof. Pavel Vitliemov, PhD,
Assoc. Prof. Mimi Kornazheva, PhD,
Boryana Stancheva, PhD,
Prof. Vladimir Chukov, DESc,
Pr. Assist. Krasimir Koev, PhD,
Prof. Juliana Popova, PhD,
Pr. Assist. Hristina Sokolova, PhD,
Pr. Assist. Magdalena Andreeva, PhD,
Assoc. Prof. Emilia Velikova, PhD,
Assoc. Prof. Bagryana Ilieva, PhD,
Pr. Assist. Reneta Zlateva, PhD,
Pr. Assist. Velislava Doneva, PhD,
Assoc. Prof. Stefka Mindova,
Assoc. Prof. Sasho Nunev, PhD,
Assoc. Prof. Despina Georgieva, PhD,
Pr. Assist. Vanya Panteleeva, PhD,
Assoc. Prof. Emil Trifonov, PhD,
Assoc. Prof. Galina Lecheva;
Assist. Prof. Milen Sapundzhiev, PhD;
Assoc. Prof. Tsvetan Dimitrov, PhD,
Assoc. Prof. Nastya Ivanova, PhD.

◆ **REVIEWERS:**

Prof. Ivan Dimov, DSc

Prof. Stepan Terzian, DSc

Assoc. Prof. Desislava Atanasova, PhD

Assoc. Prof. Evelina Veleva, PhD

Assoc. Prof. Iliya Brayanov, PhD

Assoc. Prof. Juri Kandilarov, PhD

Assoc. Prof. Miglena Koleva, DSc,

Assoc. Prof. Plamen Yalamov, PhD,

Assoc. Prof. Rumen Rusev, PhD,

Assist. Prof. Magdalena Andreeva, PhD

Financial Mathematics & Informatics

Content

1. WED-2.203-SSS-FM-01 An Estimate of Investments Portfolio's Value <i>Hristiyan Velikov, Velizar Pavlov</i>	10
2. WED-2.203-SSS-FM-02 Retrospection of Some Random Events and Approximations in Mathematics <i>Aleksandra Klimenko, Velizar Pavlov</i>	16
3. WED-2.203-SSS-FM-03 Monte Carlo Methods and Their Applications <i>Daniela Petrova, Iliyana Raeva</i>	22
4. WED-2.203-SSS-FM-04 Option Pricing Using Monte Carlo Methods <i>Boyan Boyanov</i>	28
5. WED-2.203-SSS-FM-05 Model of Liquidity Analysis with Excel <i>Byulent Idirizov, Lyubomir Lyubenov</i>	34
6. WED-2.203-SSS-FM-06 Numerical Computation of the Implied Volatility Surface from Real-Time Market Data Using Leland Model <i>Slavi Georgiev, Beáta Stehlíková</i>	41
7. WED-2.203-SSS-FM-07 Accurate and Numerical Solutions for Black-Scholes European Options <i>Nikola Borislavov, Lubin Vulkov</i>	47
8. WED-2.203-SSS-FM-08 Application of the Black-Scholes Model for Evaluating Options <i>Dimitar Stanev, Tihomir Gyulov</i>	54
9. WED-2.203-SSS-FM-09 Computation of Value-at-Risk, Expected Shortfall and Minimum Capital Requirement for Market Risk for a Multiple Asset Portfolio .. <i>Slavi Georgiev</i>	59
10. THU-1.416-SSS-I-01 Termohub – IoT Web Service <i>Borislav Kosharov, Katalina Grigorova</i>	67
11. THU-1.416-SSS-I-02 Web-based Transport Management System <i>Tohomir Vassilev, Magdalena Andreeva</i>	72
12. THU-1.416-SSS-I-03 Methodology & Techniques Used in the Development of Graphics Library with OpenGL <i>Svetlozar Iliev, Tsvetomir Vasilev</i>	78
13. THU-1.416-SSS-I-04 Bypassing Antiviruses in 2018	85

Svetoslav Hadziivanov

- | | | |
|-----|---|-----|
| 14. | THU-1.416-SSS-I-05 | 97 |
| | Treasureland – 2D platform Game | |
| | <i>Iskren Ivanov, Katalina Grigorova</i> | |
| 15. | THU-1.416-SSS-I-06 | 103 |
| | Techniques for Analysis of the Reliability of the Software Systems and Problems for Ensuring the Reliability | |
| | <i>Kamelia Dobрева, Plamenka Hristova</i> | |
| 16. | THU-1.416-SSS-I-07 | 109 |
| | Using the Methods of Clustering and Classifications for Project Task Execution Time Prediction | |
| | <i>Natalia Bobrovskaya, Pavel Akhmetchanov</i> | |
| 17. | THU-1.416-SSS-I-08 | 117 |
| | Jewelry Design Studio Database Development | |
| | <i>Yuliya Vlasova, Elena Malysheva</i> | |

AN ESTIMATE OF INVESTMENT PORTFOLIO'S VALUE ¹

Hristiyan Velikov – Student

Financial Mathematics

University of Ruse “Angel Kanchev”

E-mail: hristiyanvel@gmail.com

Prof. Velizar Pavlov, PhD

Head of Department of Applied Mathematics & Statistics,

University of Ruse “Angel Kanchev”

E-mail: vpavlov@uni-ruse.bg

Abstract: An Estimate of Investment Portfolio's Value: Investment and risk management problems are fundamental problems for both financial institutions and individual investors. Making a good analysis of one's portfolio is key to successful investing. This paper considers two possible investments. We estimate the risk premium and the investment portfolio's value. Then we suggest a way of limiting the risk through diversification and upon our analysis, show a 5-step algorithm for choosing and evaluating one's portfolio.

Key words: risk, portfolio, investment, optimization, diversification, hedging

ВЪВЕДЕНИЕ

Финансовата инвестиция е дългосрочно влягане на парични средства или капитал за придобиване на финансови инструменти или други активи с цел да се постигнат печеливши възвръщания във формата на лихва, дивиденди, повишаване на стойността на инструмента или друг вид доход (виж [1]).

Съществуват два вида инвеститори: индивидуални и институционални. И при двата вида е нужно да се мине през верига от съображения, анализи и действия преди да се достигне до реалното влягане на парични средства или така нареченото инвестиране. Всеки един инвеститор трябва да има ясно определени за него цели и ограничения. При индивидуалните инвеститори тези цели и ограничения до голяма степен се определят от това, на кой етап в жизнения си цикъл се намират. Целите на млад, току-що завършил студент, например, крайно се различават от тези на пенсионирана вдовица в преклонна възраст. От друга страна, ограниченията на институционалните инвеститори често биват усложнявани от редица правни задължения и регулиращи разпоредби.

Друга важна задача в инвестиционния процес е анализът на ценните книжа и пазара, при който оценяваме риска и очакваната възвръщаемост като атрибути на целия набор от възможните инвестиционни средства. Част от анализа на един инвеститор е определянето на най-добрите отношения риск-възвръщаемост и избора на най-добрия портфейл от множеството на възможните портфейли. По друг начин казано, съставяне на оптимален портфейл от активи.

Съществуват три основни принципа в съставянето на един портфейл, като и в трите рискът заемат централно място:

- Инвеститорите избягват риска и изискват възнаграждение, за да предприемат рискови инвестиции. Възнаграждението се приема за рискова премия – очаквана норма на възвръщаемост, която е по-висока от тази при алтернативните безрискови инвестиции;
- Всеки инвеститор трябва да направи оценка на своя личен компромисен избор между портфейлния риск и очакваната възвръщаемост. За целта използва функция на полезност,

¹ Докладът е представен на студентската научна сесия на 23. 05. 2018 г. в секция „Финансова математика“ с оригинално заглавие на български език: ОЦЕНКА НА ПОЛЕЗНОСТТА НА ИНВЕСТИЦИОНЕН ПОРТФЕЙЛ.

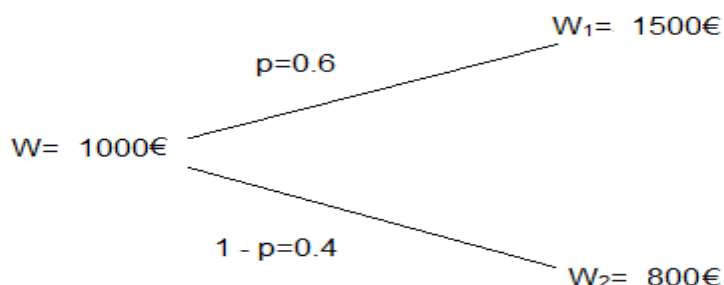
която предполага, че инвеститорите могат да присвоят стойност за благополучие или полезност на всеки инвестиционен портфейл в зависимост от неговите риск и възвръщаемост;

- Рискът на един актив не бива да се оценява отделно от портфейла, в който той е включен. Правилният начин за оценка на риска, свързан с даден индивидуален актив е да се оцени неговото влияние върху летливостта на целия портфейл от инвестиции.

РИСКОВА ПРЕМИЯ

Когато говорим за наличието на риск означава, че за дадено събитие съществува повече от един изход. В началото на нашия доклад е полезно да въведем и изясним някои основни понятия като проста перспектива.

Нека допуснем, че даден инвеститор има желание да инвестира в компания X сума $W = 1000$ €, при което в края на периода са възможни два изхода. Първият изход е благоприятен, той донася на инвеститора сума $W_1 = 1500$ € и нека вероятността за неговото настъпване е $p = 0.6$. Вторият изход е неблагоприятен, сумата е $W_2 = 800$ € и той настъпва с вероятност $1 - p = 0.4$. Тази проста перспектива можем да онагледим с дърво на събитията:



Възниква въпросът, как този портфейл може да бъде оценен. Един от начините е да се използва средствата на дискриптивната статистика и да се пресметне средната стойност или очакваната сума в края на периода $E(W)$:

$$E(W) = pW_1 + (1 - p)W_2$$

$$E(W) = 0.6 * 1500 + 0.4 * 800 = 1220,$$

което означава, че очакваната печалба е $P = 1220 - 1000 = 220$ €.

Сега можем да пресметнем дисперсията и стандартното отклонение на дадения портфейл. За тази цел използваме формулата за дисперсия:

$$\sigma^2 = p[W_1 - E(W)]^2 + (1 - p)[W_2 - E(W)]^2$$

$$\sigma^2 = 0.6(1500 - 1220)^2 + 0.4(800 - 1220)^2 = 117600,$$

откъдето $\sigma = 342.93$.

Следователно тази инвестиция е доста рискова, тъй като стандартното отклонение на печалбата е много по-голямо от очакваната печалба 220 €. За да има реална представа дали очакваната печалба е достатъчно голяма, за да оправдае такъв риск, инвеститора е нужно да проучи алтернативни портфейли.

Нека допуснем, че съкровищните бонове са една такава алтернатива, при което за същия период възвръщаемостта е 4%, т.е. при инвестирани 1000 €, те биха донесли сигурна печалба от 40 €.

Така се очертават две възможности: проста перспектива с очаквана печалба 220 € и инвестиция в съкровищни бонове със сигурна печалба 40 €. Следователно очакваната пределна печалба на рисковия портфейл, спрямо инвестицията в безопасни съкровищни

бонове е 180 €. Тази пределна печалба се нарича още **рискова премия**, която се явява компенсация за инвестиционния риск.

Рисковата премия в този случай е положителна, което означава, че инвеститорът може да приеме дадената проста перспектива или да я отхвърли ако допълнителната печалба (180 €) не е достатъчно голяма, за да компенсира поемания риск.

РИСК-ВЪЗВРАЩАЕМОСТ

Инвеститорите, които избягват риска не приемат инвестиционни портфейли, които имат нулева или отрицателна рискова премия. Още повече, инвеститорът-рискофоб „глобява“ с известен процент очакваната норма на възвръщаемост от даден рисков портфейл, за да се отчете свързания с портфейла риск. Колкото по-голям е рискът, възприеман от инвеститора, толкова по-голяма е глобата.

Тази система за риск-глоба обикновено се използва от финансовите теоретици и се нарича оценка/стойност на полезност. Всеки инвеститор може да направи такава оценка на конкуриращите се инвестиционни портфейли в зависимост от очакваните възвръщаемост и риск на тези портфейли. Тази оценка за полезност/благополучие често бива спомената като средство за класиране на портфейли, като по-високи оценки получават портфейлите с по-привлекателни профили риск-възвръщаемост. Формулата, която се използва е:

$$U = E(W) - 0.005 * A * \sigma^2 \quad (1)$$

където U е стойността на полезност, а A е индексът на рискофобия на инвеститора, който се изменя в граници от 1 до 8, като колкото по-голяма е стойността на A , толкова по-склонни са инвеститорите да избягват риска.

В разглеждания случай инвеститорът имат очаквана възвръщаемост 22% и стандартно отклонение 34%. Ако допуснем една умерена стойност на параметъра за избягване на риска $A = 3$, то за оценката на полезност на портфейла по формула (1) се получава:

$$U = 22 - 0.005 * 3 * 34^2 = 4.66\%,$$

което е малко повече от безрисковия процент в случая на инвестиране в съкровищни бонове.

На тази база се въвежда понятието "еквивалентен сигурен процент" (certainty equivalent rate) на даден портфейл, който е равен на процентната доходност, която би трябвало да предлагат безрисковите инвестиции, за да са еднакво привлекателни с рисковия портфейл (виж [1]).

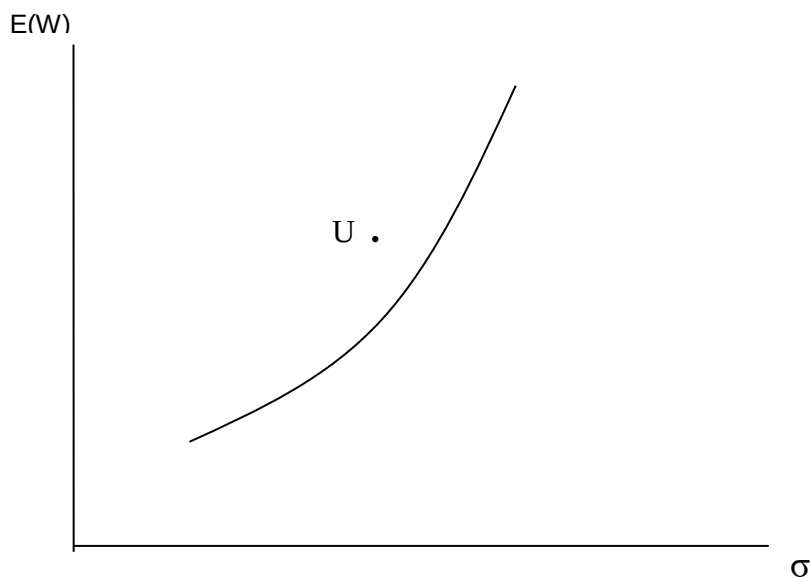
Нека допуснем, че имаме няколко различни портфейла - с висока възвръщаемост и висок риск, както и с по-нисък риск, но и по-ниска възвръщаемост (виж таблица 1).

Таблица 1. Оценка на полезност

$E(W) \%$	$\sigma \%$	$U \%$
10%	20.00%	$10 - 0.005 * 3 * 400 = 4\%$
15%	27.08%	$15 - 0.005 * 3 * 733.32 = 4\%$
20%	32.65%	$20 - 0.005 * 3 * 1066.02 = 4\%$
25%	37.40%	$25 - 0.005 * 3 * 1398.76 = 4\%$

Инвеститорите (от тип A=3) ще бъдат еднакво привлечени от всички горе споменати портфейли, защото имат еднаква оценка на полезност. Тези портфейли ще лежат върху една крива в графиката възвръщаемост-отклонение, която свързва всички портфейли с една и съща стойност на полезност (фигура 1), наречена **крива на безразличие** (виж [3]).

Всеки портфейл намиращ се на северозапад от кривата на безразличие е по-добър от тези, лежащи върху кривата или на югоизток от нея, които са по-непривлекателни. Разглежданият портфейл има оценка на полезност $U = 4.66\%$, която се намира над кривата на безразличие, което го прави добър (фигура 1).



Фигура 1. Крива на безразличие и оценка на полезност

ОГРАНИЧАВАНЕ НА РИСКА

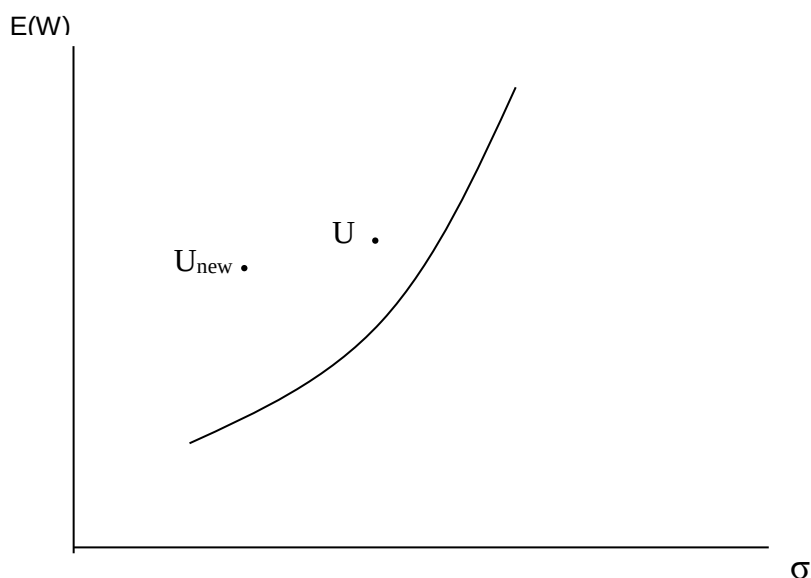
Друга важна задача, свързана с полезността на портфейла е ограничаването на риска. Това може да се направи чрез диверсификация (виж [2]).

В разглеждания случай нека предположим, че инвеститорът решава да разпредели парите си наполовина между сигурните съкровищни бонове (TB - treasury bills) и рисковия портфейл. Тогава новата очаквана норма на възвръщаемост е претеглена средна стойност на очакваните две норми:

$$E(W_{new}) = 0.5E(W) + 0.5E(TB) = 0.5 * 22 + 0.5 * 4 = 13\%$$

Естествено новият портфейл има и ново претеглено стандартно отклонение:

$$\sigma_{new} = 0.5\sigma = 0.5 * 34 = 17\%$$



Фигура 2. Нова оценка на полезност спрямо крива на безразличие

Лесно се вижда, че както очакваната възвръщаемост, така и стандартното отклонение имат по-малки стойности. По формула (1) за оценката на полезност получаваме

$$U_{new} = 13 - 0.005 * 3 * 17^2 = 8.67\%$$

Ако сравним първата оценка на полезност и тази на комбинирания портфейл от рисков актив и съкровищни бонове, виждаме, че тя е почти удвоена.

Графичната интерпретация (фигура 2) показва, че новата оценка на полезност U_{new} се намира по-на северозапад от кривата на безразличие в сравнение с U , което означава, че новият комплексен портфейл е по-добър, макар и очакваната възвръщаемост да е по-малка.

ЗАКЛЮЧЕНИЕ

Извършване на предварителен анализ е от изключителна важност при съставянето на един инвестиционен портфейл. Един добър инвеститор не си набавя информацията от вестници, усещания или слухове [4]. За да бъде успешен и печеливш, инвеститорият трябва да проучи пазара и да направи набор от потенциални инвестиции в рискови активи или съкровищни бонове. След това да оцени всички алтернативни портфейли, за да има реална представа къде по графиката възвръщаемост – отклонение се намира избрания портфейл. Едва тогава, той може да вземе съответното инвестиционно решение: инвестиция с ниска възвръщаемост и нисък риск или такава с висока възвръщаемост, но и съответно по-висок риск.

Въз основа на направения анализ бихме могли да предложим алгоритъм за избор на инвестиционен портфейл, включващ следните стъпки:

1. Намиране на средната очаквана възвръщаемост на даден актив спрямо исторически данни за актива.
2. Пресмятане на стандартното отклонение относно вече изчислената очаквана възвръщаемост.
3. Намиране на оценка на полезност на актива и визуализиране върху графиката възвръщаемост – отклонение.
4. Диверсификация на риска - добавяне на актив / съкровищни бонове към портфейла и произчисляване на т. 1, 2 и 3.
5. Съпоставка на двата портфейла спрямо кривата на безразличие.

При изпълнението на тези стъпки се очаква един инвеститор да има по-добра представа за полезността на своя портфейл. При наличието на много такива алтернативни портфейли, той може да вземе най-доброто решение.

REFERENCES

- [1] Z. Bodie, A. Kane, A. Marcus, *Investments*
- [2] Keith Pilbeam, *Portfolio Analysis: Risk and Return in Financial Markets*
- [3] Hult, H., Lindskog, F., Hammarlid, O., Rehn, C.J., *Risk and Portfolio Analysis*
- [4] A. Vichev, *An Introduction to Investing seminar*

RETROSPECTION OF SOME RANDOM EVENTS AND APPROXIMATIONS IN MATHEMATICS ²

Aleksandra Klimenko – Student

Financial Mathematics

University of Ruse “Angel Kanchev”

E-mail: alexklimenko@abv.bg

Prof. Velizar Pavlov, PhD

Head of Department of Applied Mathematics & Statistics,

University of Ruse “Angel Kanchev”

E-mail: vpavlov@uni-ruse.bg

Abstract : Retrospection of Some Random Events and Approximations in Mathematics: Mathematics is one of the most significant and at the same time topical achievements of the humanity. It is a part of any scientific discipline and finds varied applications in a number of areas of everyday life. Going back in time, we learn not only who are the founders of the most significant discoveries in mathematics, but also interesting facts. It is important to know them because of general knowledge. This paper presents a different view to some random events and approximations, showing mathematics used in practice. In fact, they are the fundamentals of financial mathematics. Thanks to random events and approximations we can evaluate insurance risk opportunities, risk-profitability dependencies and predict anomalies of the investment's market.

Key words: retrospection, mathematics, random events, approximations

ВЪВЕДЕНИЕ

Великият швейцарски математик, физик и философ Леонард Ойлер казва: "Математиката е ключът на цялото човешко знание". Чрез нея можем да обясним цветовете на залеза или структурата на мозъка, можем да вникнем в субатомния свят или в галактиките на далечния Космос. Според друг велик изследовател Леонардо да Винчи: "Никаква достоверност няма в науките там, където не може да се приложи математиката и в това, което няма връзка с математиката".

Сложната математическа теория се основава на математически формули, в които може да види толкова много красота, колкото има в най-прекрасната музика на великите композитори или в шедьоврите на изобразителното изкуство. Тази теория става особено полезна, когато се прилага в практиката (виж [8]). Тази задача може да бъде решавана успешно само от хора, които имат нужната теоретична подготовка и умения да направят тази важна връзка, а процесът на нейното решаване винаги съдържа риск за грешка. Причината за този риск е най-общо, че в практиката винаги влияят и случайни фактори, които не могат да бъдат предвидени.

В настоящата работа е представен един по-различен поглед към някои случайности и приближения, изучавани от математиката и така нужни за практиката.

ЗАРОВЕ

Заровите са едни от най-ранните методи за генериране на случайни числа (фиг. 1). Първоначално били изработвани от глезенната кост на домашните животни, като агнетата. В древността хората вярвали, че заровите могат да предсказват бъдещето, тъй като се смятало, че самите богове контролират изхода от хвърлянията.

² Докладът е представен на студентската научна сесия на 23. 05. 2018 г. в секция „Финансова математика“ с оригинално заглавие на български език: РЕТРОСПЕКЦИЯ НА НЯКОИ СЛУЧАЙНОСТИ И ПРИБЛИЖЕНИЯ В МАТЕМАТИКАТА

В продължение на векове хвърлянията на зар се използват за преподаване на теория на вероятностите (виж [5]). Ако разгледаме стандартен зар с 6 страни, върху всяка от които има различно число, то е добре известно, че вероятността да се падне едно конкретно число при едно хвърляне е $1/6$. Вероятността да се паднат серия от i числа в определена поредица е $1/6^i$.



Фигура 1. Зарове



Фигура 2. Колело на каруца

ЧИСЛОТО π

Добре известно е, че числото π е приблизително равно на 3,14 и представлява отношението между дължината на дадена окръжност и нейния диаметър. Навярно древните хора са го забелязали благодарение на това, че при всяко завъртане на колелата на каруца тя се придвижва напред с около три пъти диаметъра на колелото (фиг. 2). През 1706 г. уелският математик Уилям Джоунс въвежда символа π , който е първата буква на гръцката дума за периферия (виж [6]). Още по-рано, през 1673 г., немският математик Готфрид Вилхелм Лайбниц (виж [9]) открива формулата, чрез която π може да бъде представено по следния начин:

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots$$

ЛОГАРИТМИ

Джон Непер, откривателят на логаритмите, създава устройство за пресмятане, известно като "плочки на Непер" (фиг. 3). Това са пръчици, изписани с таблици на умножението, които могат да се подреждат в определен ред и да улесняват пресмятанията. Въртящите се пръчици позволяват умножението да се представи като поредица от просто събиране. Логаритъмът на числото x при основа a е равен на b и се изписва като

$$\log_a x = b \leftrightarrow x = a^b$$



Фигура 3. Плочки на Непер



Фигура 4. Числото нула

НЕПЕРОВО ЧИСЛО

Якоб Бернули открива числото през 1685 г. при изучаване на сложната лихва. Ако предположим, че в банкова сметка имаме 1 лв. при годишна лихва 100%, при което сметката се олихвява веднъж годишно, то след изтичане на годината ще имаме 2 лева. Ако лихвата се начислява веднъж на 6 месеца от годината (по 50% на шестмесечие), то в края на годината ще получим 2,25 лв. Ако лихвата се начислява ежемесечно ($100\% : 12 = 8,33\%$ на месец), то в края

на годината ще имаме 2,61 лв. Ако лихвата се начислява ежедневно ($100\% : 365 = 0,274\%$ на ден), то в края на годината ще имаме 2,71 лв. Така колкото по-често се олихвява сметката, толкова по-голяма сума се получава. Обаче крайният резултат не расте неограничено. Бернули забелязва, че резултатът не надхвърля определена граница, а именно 2,72 лв., към която стойност се приближава паричната сума в края на годината, когато сметката се олихвява все по-често и по-често. Той доказва, че тази гранична стойност е числото $2,718281828459045\dots$, което по-късно е наречено Неперово число в чест на Джон Непер - изобретателя на логаритмите, при което $\log_e x = \ln x$ (виж [7]).

НУЛА

Първоначално древните вавилонци нямали символ за нула. Това води до несигурност в техните означения. Вавилонските писари оставяли единствено празен интервал там, където нулата трябвало да се постави. Ето защо в средата или в края на числата не било никак лесно да се различи броят на празните места (например 12, 102 или 1002). Около 650 г. от н.е. индийските математици Брахмагупта, Махавира и Бхаскара използвали нулата в математически операции. Брахмагупта обяснява, че число, от което бива извадено същото число, прави нула. По-късно математикът Хюсеин Аршам (виж [1]) пише: „Въвеждането на нулата в десетичната бройна система през XIII в. е най-значимото постижение в развитието на система от числа, чрез която пресмятането на големи числа става възможно. Без нея описанието на процесите в търговията, астрономията, физиката, химията и индустрията би било немислимо. Липсата на такъв символ е един от сериозните недостатъци на римската бройна система.” Така въвеждането на нулата възпламенява истински огън, който в крайна сметка позволява на човечеството да работи по-лесно с големи числа и да стане ефективно в изчисленията (фигура 4).

ФОРМУЛА НА СТИРЛИНГ

Добре известно е, че за положителни цели числа n с $n!$ се означава произведението на всички положителни цели числа, които са по-малки или равни на n . Това означение е въведено от френския математик Кристиан Крамп през 1808 г. Факториелите са съществени за комбинаториката, където е важно да се познава броя възможни начини, по които могат да се подредят обектите в дадена редица. Те се срещат и в теория на вероятностите, теория на числата и математическия анализ (виж [2]).

През 1730 г. шотландският математик Джеймс Стирлинг в книгата си "Methodus Differentialis" представя приближението

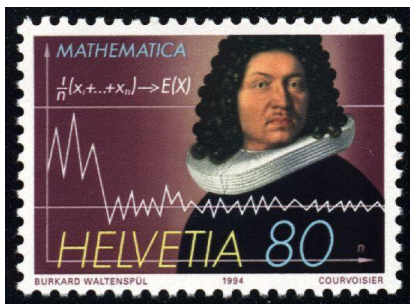
$$n! \approx \sqrt{2\pi} e^{-n} n^{n+\frac{1}{2}}$$

ЗАКОН ЗА ГОЛЕМИТЕ ЧИСЛА

Законът за големите числа (ЗГЧ) е теорема от Теория на вероятностите, доказана от швейцарския математик Якоб Бернули [3]. Тя описва дългосрочната стабилност на произволна величина. Например, когато броят наблюдения на даден експеримент (като хвърляне на монета) е достатъчно голям, то процентът от случаите, в които се наблюдава даден определен резултат спрямо всички опити (например да се пада ези), ще се доближава до вероятността за този резултат (в случая 0,5). Застрахователите използват ЗГЧ, за да направят оценка на рисковете и загубите.

Бернули пресмята разпределението на белите топки в урна, пълна с неизвестен брой черни и бели топки. Изтеглянето на топки от урната и случайната замяна на топка при всяко изтегляне позволява на Бернули да изчисли съотношението на белите спрямо всички изтеглени топки. Бернули открива, че ако направи това достатъчно голям брой пъти, може да получи желаната точност за търсения резултат и прави извода, че "Ако наблюденията на всички събития продължат до безкрайност, то вероятността започва да клони към пълна сигурност, всичко на света би се случвало, дори и най-малко вероятните събития ще трябва да се случат при определени обстоятелства". В чест на Бернули на швейцарска пощенска марка

от 1994 г. е изобразен неговият образ и е представено изображение на графиката и формулата на ЗГЧ (фиг. 7).



Фигура 7. Швейцарската марка от 1994 г.



Фигура 8. Банкнота от 10 германски марки

НОРМАЛНО РАЗПРЕДЕЛЕНИЕ

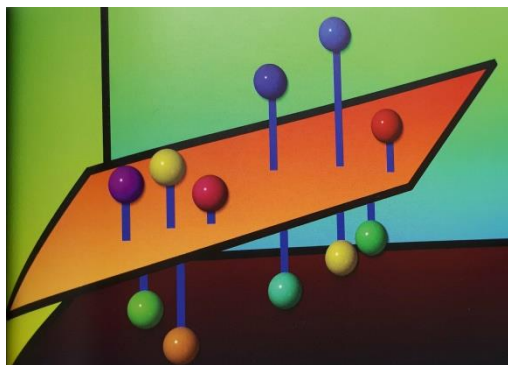
През 1733 г. френският математик Абрахам дьо Муавър за първи път описва кривата на нормалното разпределение. Известно е и като „Гаусово” в чест на Карл Фридрих Гаус, който изучавал кривата години по-късно. Тя представлява важно семейство от непрекъснати вероятностни разпределения, приложими към всички научни дисциплини, където се извършват експерименти (виж [3]).

Нормалното разпределение се определя от средната стойност и от стандартното отклонение, което определя доколко измерената стойност се отклонява от истинската. Представено графично, разпределението описва симетрична камбановидна крива. В чест на Гаус до 2002 г. на банкнотата от 10 германски марки е изобразен неговият образ и кривата на нормалното разпределение (Фиг. 8).

МЕТОД НА НАЙ-МАЛКИТЕ КВАДРАТИ

Първата значима монография, която комбинира теория на вероятностите с математическия анализ, е "Аналитична теория на вероятностите" на френския математик и астроном Пиер-Симон Лаплас [4]. В нея той разглежда методи за определяне на вероятността за случване на дадени съставни събития на базата на вероятността на отделните компоненти. В книгата са описани методът на най-малките квадрати и на т. нар. "игла на Бюфон".

Методът на най-малките квадрати днес е част от повечето статистически компютърни програми, които чертаят линии или плавни извивки през необработени експериментални данни. Това е математическа процедура за откриване на "най-добрата" крива, описваща даден набор от данни. Построява се като се минимизира сумата от квадратите на отклоненията на точките с данни спрямо тази крива. Интерпретация в тримерния случай е представена на фиг. 9.



Фигура 9. Метод на най-малките квадрати.



Фигура. 10 Граф Дьо Бюфон (1727-1775)

ЗАДАЧА ЗА ИГЛАТА НА БЮФОН

Монте Карло е административен квартал в Монако, известен с многото си казина. Но методите „Монте Карло”, кръстени на него, са основни за математиката и науката. Те използват случайността, за да решават задачи, вариращи от областта на статистиката до

верижните ядрени реакции и управлението на светофарите по пътищата. Едно от най-ранните и известни приложения на метода е през XVIII в. Тогава френският естественик и математик граф Дьо Бюфон (фиг. 10) пуска игла многократно върху лист хартия с нарисувани линии и отчита броят пъти, в които иглата докосва линия. По този начин той получава стойността на математическата константа π .

Днес свръхмощни компютри могат да генерират огромен брой псевдослучайни числа в секунда. По този начин учените могат да се възползват напълно от методите „Монте Карло” и да изучават проблеми от областта на икономиката, физиката, химията, структурата на протеините, образуването на галактиките.

ХИ-КВАДРАТ

Тестът на хи-квадрата е представен за пръв път през 1900 г. от британския математик Карл Пиърсън. Оттогава до сега неговият метод се използва в безброй различни сфери: от криптография и сигурност до анализ на резултатите на играчите в бейзбола. При прилагането на теста се приема, че резултатите са независими. Стойността на хи-квадрат може да бъде изчислена, ако знаем всяка наблюдавана честота f_e и теоретична (очаквана) честота f_t за дадено събитие по формулата

$$\chi^2 = \sum \frac{(f_e - f_t)^2}{f_t}$$

Ако честотата на очакваните и наблюдаваните събития е еднаква, то тогава $\chi^2 = 0$. Нека например тестваме хипотеза, която гласи, че популацията при бръмбари и пеперуди е еднаква (фигура 11). За целта е направена случайна извадка от 100 произволни насекоми, при което се оказва, че в нея имаме 10 бръмбара и 90 пеперуди. Тогава имайки предвид, че очакваните честоти са по 50 броя от всеки вид, то за χ^2 получаваме

$$\chi^2 = \frac{(10 - 50)^2}{50} + \frac{(90 - 50)^2}{50} = 64$$



Фигура 11. Популация от пеперуди и бръмбари

Получената стойност 64 се различава съществено от нулата, което подсказва, че началната хипотеза не е вярна.

МЕДАЛ НА ФИЛДС

Филдсовият медал е най-известната и влиятелна награда в математиката. Както и Нобеловата награда за постижения в други области на науките, Филдсовият медал е учреден от желанието математиката да се издигне над националните вражди. Той се присъжда на всеки четири години и е награда за минали постижения и стимул за бъдещи проучвания.

Понякога наричат Филдсовия медал „Нобелова награда за математика”. Той обаче се връчва само на математици на възраст до 40 години. Паричната сума е сравнително малка.

През 2006 г. е едва 13500 долара в сравнение с Нобеловата награда, която е над милион долара. Наградата е учредена от канадския математик Джон Чарлз Филдс и е връчена за първи път през 1936 г.

На лицевата страна на медала е изобразен гръцкият учен Архимед, а фразата на латински на обратната страна означава „Този медал се дава на математици, събрани от цял свят, заради изключителна работа” (фиг. 12).



Фигура 12. Медалът на Филдс

Любопитно е, че около 25 процента от медалистите са евреи, а почти половината са работили в Института за напреднали проучвания в Принстън, Ню Джърси.

ЗАКЛЮЧЕНИЕ

Бърtrand Ръсел пише: „Математиката, ако я разглеждате правилно, притежава и показва не само истина, но и върховна красота – сурова и студена красота, подобна на изящна скулптура.”

REFERENCES

- [1] Arsham, Hossein, „Zero in Four Dimensions: Historical, Psychological, Cultural, and Logical Perspectives”, The Pantaneto Forum, 2(5), 2002
- [2] Ball, Keith M., *Strange Curves, Counting Rabbits, and Other Mathematical Explorations*, Princeton, New Jersey: Princeton University Press, 2003
- [3] Galton, Francis, *Natural Inheritance*, London: Macmillan, 1889
- [4] Gordon, J. et al., „Superconducting-Normal Phase Boundary of a Fractal Network in a Magnetic Field”, Physical Review Letters 56(21): 2280-2283, 1986
- [5] Hayes, Brian, „Randomness as a Resource”, American Scientist 89(4): 300-304, 2001
- [6] Jones, William, „A New Introduction to the Mathematics”, J. Matthews, 1706 пи
- [7] Kasner, Edward and James Newman, *Mathematics and the Imagination*, New York: Dover, 2001
- [8] Pickover, Clifford A., *The Math Book*, Sterling Publishing Company, Inc., 2009
- [9] Roy, Ranjan, „The discovery of the series formula for π by Leibniz, Gregory, and Nilakantha”, Mathematics Magazine, 63: 291-306, 1990

MONTE CARLO METHODS AND THEIR APPLICATIONS³

Daniela Petrova – Student

Financial Mathematics

University of Ruse “Angel Kanchev”

E-mail: danityyyyy@gmail.com

Assoc. Prof. Iliyana Raeva, PhD

Department of Applied Mathematics and Statistics

University of Ruse “Angel Kanchev”

E-mail: iraeva@uni-ruse.bg

Abstract: Monte Carlo Methods are a broad class of computational algorithms, which are applied for finding the solution of a problem, which has multiple set of input data. Monte Carlo Methods simulate the random movement of prices of the financial assets and are also used for obtain estimates of options logical statements and concepts from probabilities and statistics such as central limit theorem, mathematical expectation, confidence intervals and others. It is important that the solution, found by Monte Carlo Methods, is with very high accuracy.

Keywords: Monte Carlo Methods, algorithms, random movement, financial assets, solution with high accuracy

ВЪВЕДЕНИЕ

Методите Монте Карло са широк клас от изчислителни алгоритми, които се прилагат за намиране на решение на посочен проблем, който има множество зададени входни данни. Ранният вариант на метода може да бъде видян в експеримента „Иглата на Буфон“, проведен още през XVIII в. Той представлява предположение за под, направен от паралелни и равни по ширина дървени дъски, като чрез Методи Монте Карло се изчислява каква е вероятността, ако бъде изпусната игла на пода, тя да попадне така, че да пресича разстоянието между две дъски [3].

Модерната версия на Марковата верига на Методи Монте Карло е изработена от Станислав Улам през 40-те години на XX в. Той работил по проекти в националната лаборатория „Лос Аламос“. Научавайки за работата му, Джон фон Нойман създава електронноизчислителна машина (ENIAC), чрез която прилага изчисленията на Методите Монте Карло. Името Методи Монте Карло произлиза от съвместната работа на фон Нойман и Улам, която изисквала кодово име. Тогава техен колега, Николай Метрополис, предложил името Монте Карло, на името на казиното Монте Карло в Монако, където чичото на Улам взема пари назаем от роднини, за да залага [2].

Методите Монте Карло също така са били много важни за проекта „Манхатън“ (проект за създаване на първите ядрени оръжия, осъществен в САЩ между 1939 и 1946г.) [3].

ИЗЛОЖЕНИЕ

Структурата на Методи Монте Карло зависи единствено от началната стойност S_0 на цената на разглеждания актив, т.е. цената S в началния момент t_0 , от който започва опцията. Методът на Монте Карло симулира случайното движение на цените на финансовите активи и използва за оценяването на опциите логически твърдения и понятия от вероятностите и статистиката като централна гранична теорема, математическо очакване, доверителни интервали и др. [1].

Като начало се симулират голям брой ценови пътеки, описващи случайното движение на дадения ценови процес S_t в интервала $[0, T]$. Пресмята се функцията на печалба и

³ Докладът е представен на студентската научна сесия на 23. 05. 2018 г. в секция „Финансова математика“ с оригинално заглавие на български език: МЕТОДИ МОНТЕ КАРЛО И ПРИЛОЖЕНИЯ.

математическото очакване за всяка от примерните ценови пътеки, след което се пресмята премиумът на дадената опция чрез дисконтиране и намирането на доверителен интервал за средната стойност от всички симулирани ценови пътеки.

При оценяване на Европейски опции е нужно да се намери математическото очакване на функцията на печалба само в датата на падежа T . Полученият резултат в момент T трябва да се дисконтира с множител e^{-rT} , за да получим премиума на дадената опция в настоящия момент $t=0$, т.е:

$$V(S_0, 0) = e^{-rT} \hat{E} (V(S_T, T)). \quad (1)$$

За да приложим метода Монте Карло към модела на Блек-Шолс $\frac{dS}{S} = rdt + \sigma dW_t$ за пресмятане на премиума на Европейска кол опция, пресмятаме функцията на печалба:

$$V(S, T) := \max \left[S_0 e^{\left(r - \frac{1}{2}\sigma^2\right)T + \sigma\epsilon\sqrt{T}}, 0 \right], \quad (2)$$

където K е страйкова цена, S_0 – цена на разглеждания актив, лихва r и волатилност σ са предварително определени от банката и фиксирани през целия живот на опцията $t \in [0, T]$.

Премиумите на опцията $V(S_T, 0)$ за всяка ценова пътека, получена по горната формула, се получават чрез дисконтиране на функцията на печалба с множителя e^{-rT} :

$$V(S_0, 0) := e^{-rT} \left(\max \left[S_0 e^{\left(r - \frac{1}{2}\sigma^2\right)T + \sigma\epsilon\sqrt{T}}, 0 \right] \right). \quad (3)$$

За завършване на прилагането на метода Монте Карло се пресмята доверителен интервал за стойностите на премиумите на опцията.

За да бъде постигнато необходимото ниво на точност се използват голям брой ценови пътеки. Ако искаме да увеличим точността с една десета е нужно да увеличим стократно броя на генерираните ценови пътеки.

Евристичната структура на метода Монте Карло е основно негово предимство, тъй като при него могат лесно да се симулират ценови пътеки на нестандартни деривати. Пример за нестандартни деривати са бариерните опции, които са много популярни деривати на финансовите пазари. Съществуват две основни причини за търгуването на голямо количество бариерни опции на борсовите пазари:

1. имат по-ниски премиуми от съответните ванилови опции
2. компенсациите, които се изплащат в случай, че опцията се анулира, т.е. цената на разглеждания актив е достигнала предварително определена бариера преди датата на падеж [1].

Симулацията на случайни числа често е единственият начин да се опише и генерира вероятностното разпределение S_t на разглеждания актив.

Големината на времевите стъпки се избира така, че ценовите пътеки да стимулират реалистично стохастичния процес S_t в интервала $0 \leq t \leq T$. За тази цел се минава през следните две неща:

1. Времето $0 \leq t \leq T$ до падежа T на дадена опция се дискретизира чрез избиране на подходяща времева стъпка Δt , играеща роля на времевия интервал δt във формула

$$S_{t+\delta t} = S_t \exp \left(\left(r - \frac{1}{2}\sigma^2 \right) \delta t + \sigma \epsilon \sqrt{\delta t} \right), \text{ където } j = 1, \dots, N \text{ и } t_0 + N\Delta t = T;$$

2. Генериране на ценови пътеки S_t , всяка от които бива генерирана поотделно чрез последователни стойности $S_{t_0}, \dots, S_{t_0+j\Delta t}, \dots, S_T$, $j = 1, \dots, N$, $t_0 = 0$ и $t_0 + N\Delta t = T$ от

$$\text{формула } S_{t+\delta t} = S_t \exp \left(\left(r - \frac{1}{2}\sigma^2 \right) \delta t + \sigma \epsilon \sqrt{\delta t} \right).$$

Цената на разглеждания актив в модела на Блек-Шолс S_t има логнормално разпределение (log-normal distribution) за разлика от модела на Мертон, където цената на разглеждания актив S_t следва процес на Леви $L_t = \mu dt + \sigma W_t + \sum_{i=1}^{N_t} Y_i$, който е с по-сложно разпределение:

$$S_t = S_0 \exp\left(\mu t + \sigma W_t + \sum_{i=1}^{N_t} Y_i\right), \text{ където } \mu = p - \frac{1}{2}\sigma^2 - \lambda\left(\exp\left(\gamma + \frac{1}{2}\sigma^2\right) - 1\right) -$$

от познатото в икономиката логнормално от формула $S_{t+\delta t} = S_t \exp\left(\left(r - \frac{1}{2}\sigma^2\right)\delta t + \sigma\varepsilon\sqrt{\delta t}\right)$.

Пример 1.

Нека симулираме 30 ценови пътеки, като цената на разглеждания актив в началния момент е $S_0 = 35$ и се описва от случаен процес, следващ Брауново лутане. За симулациите са използвани следните параметри: $NRepl = 30$ е броят на симулираните пътеки, безрисковият лихвен процент е $r = 0,07$ и волатилността е $\sigma = 0,1$ за период от една година, т.е. $T = 2$ (този период може да се разглежда като падеж за даден вид опция).

Общият брой на междинните времеви моменти използвани за симулацията на ценовите пътеки е 100, т.е. имаме 50 стъпки по времето и съответно параметър $NSteps = 100$.

Цената на дадения актив за малък период от време не се изменя рязко. Това е основно отличително свойство на Брауновото лутане. Чрез тази програма са генерирани ценови пътеки, които може да видите на фигурата по-долу. Графиките потвърждават две основни отличителни характеристики на ценовите пътеки, следващи Брауново лутане в модела на Блек-Шолс:

1. непрекъснати промени (continuous changes) на стойността на ценовата пътека, т.е. цената на разглеждания актив има повишение или намаляване във всеки един момент
2. малки изменения в цената на разглеждания актив за кратък период от време, т.е. цените не могат да се променят рязко. Много е малка вероятността да се случат големим промени в цената, което изключва появата на скокове [1].

Програма 1

```
%Симулация на ценови пътеки в модела на Блек-Шолс
% function SPaths =
% Pr1(S0, r, sigma, T, NSteps, NRepl)
NRepl = 30; % брой ценови пътеки
NSteps = 100; % брой времеви стъпки
T = 2; % матуритет
S0 = 35; % начална цена на актива
r = 0.07; % безрисков лихвен процент
sigma = 0.1; % параметър на волатилност
dt=T/NSteps; % дължина на времевите стъпки
% Формула (3.3) за симулация на ценови пътеки:
%  $S_{t+\delta t} = S_t \exp\left(\left(r - \frac{1}{2}\sigma^2\right)\delta t + \sigma\varepsilon\sqrt{\delta t}\right)$ , където  $\delta t := dt$ ,  $\varepsilon \sim N(0,1)$ .

nudt=(r-0.5*sigma^2)*dt; %  $nudt := \left(r - \frac{\sigma^2}{2}\right)dt$ 

sidt=sigma*sqrt(dt); %  $sidt := \sigma\sqrt{dt}$ 
% Генериране на ценови пътеки като стойностите им
% се запазват в матрица с размерност [NRepl × NSteps],
% NRepl е броят на ценовите пътеки на
```



```
% NSteps е броят на еднаквите времеви стъпки
Increments = nudt+sidt*randn(NRepl,NSteps);
LogPaths = cumsum([log(S0)*ones(NRepl,1),Increments],2);
% Ценови пътеки в нова матрица SPaths
SPaths = exp(LogPaths);
% Цикъл за генериране на последователни времеви  $t_j$ 
% стъпки:  $t_1, t_2, \dots, t_{NSteps+1}, t_1 = 0, t_2 = dt, t_3 = 2dt, \dots, t_{NSteps+1} = (NSteps)dt$ 
for j = 1:NSteps+1
    t(j) = (j-1)*dt; % в програмата  $t(j) := t_j$ 
end
% Генериране на фигура с ценовите пътеки SPaths
plot(t, SPaths)
% Етикети за заглавие и оси на генерираната фигура
title('Симулация на ценови пътеки в модела на Блек-Шолс', 'FontSize', 12)
xlabel('Време t до падежа T, T=2')
ylabel('30 ценови пътеки генерирани от  $S_0=35$ ')
```



Фиг. 1. Симулация на ценови пътеки, моделирани чрез случаен процес: Брауново лутане в Модела на Блек-Шолс

Пример 2. Нека симулираме ценови пътеки, като движението на цената на разглеждания актив се описва от процес на Леви, използван в модела на Мертон.

За да можем да направим сравнение с модела на Блек-Шолс, ще използваме същите параметри от програма 1, т.е. безрисков лихвен процент $r = 0,07$ и волатилност $\sigma = 0,1$ за период две години $T = 2$ (този период може да се разглежда като падеж за даден вид опция), а броят на симулираните ценови пътеки е 30 и цената на разглеждания актив в началния момент $S_0 = 35$. Примерът е изследван чрез програма 2.

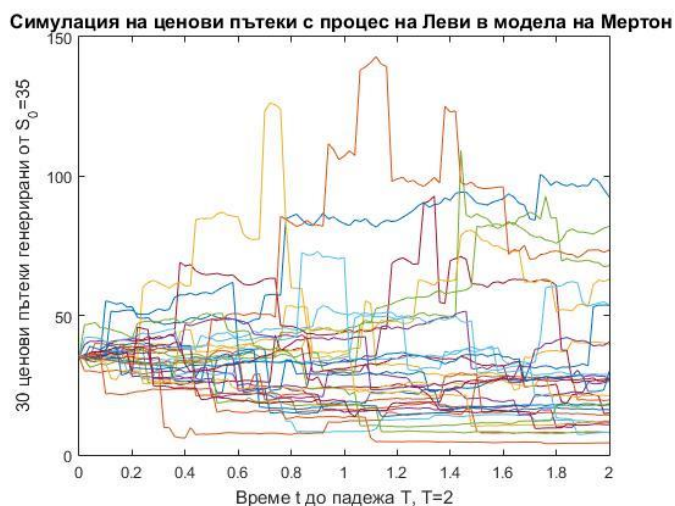
Програма 2

```
% Симулация на ценови пътеки в модела на Мертон
N = 100; % брой времеви стъпки
M = 30; % брой ценови пътеки
T = 2; % матуритет
price = 35; % начална цена на актива
strike = 35; % страйкова цена
rate = 0.07; % лихва
```

```

sigma = 0.1; % волатилност
div = 0; % дивидент
lambda = 5; % параметър за интензитета на скоковете
% Теоретични формули за разпределенията
%  $\log(1+k) \sim N(\text{Gamma1}, \text{Delta}^2)$ 
%  $\text{Gamma1} = \text{Gamma} - 0.5 * \text{Delta}^2$ ,
%  $E(k) = \text{kappa} = \exp(\text{Gamma}) - 1$ 
Gamma = 0; Delta = 0.3;
Gamma1 = Gamma - 0.5*Delta^2;
kappa = exp(Gamma) - 1;
dt = T/N; % времева стъпка
for i = 1:N+1
    t(i) = (i-1)*dt;
end
sum = 0; % Нулева стойност за сумиране на печалби
for i = 1 : M
    % Начална цена на актива за всяка симулация
    S(1,i) = price;
    for j = 2:N+1
        % Генериране на случайните числа чрез
        % стандартно Нормално разпределение
        deviate = normrnd(0,1);
        % Генериране на случайни числа чрез Поасоново
        % разпределение
        temp = poissrnd(lambda*dt);
        TJS = 0;
        for k = 1:temp
            JS = normrnd(Gamma1,Delta);
            TJS = TJS + JS;
        end
        % Генериране на ценовите пътеки S(j,i)
        S(j,i) = S(j-1,i)*exp((rate-div-lambda*kappa-
0.5*sigma^2)*dt+sigma*sqrt(dt)*deviate+TJS);
    end
    payoff = max(S(N+1,i)-strike, 0); % чиста печалба
    sum = sum + payoff; % сумиране на печалбите
end
% Цена на ванилова Европейска кол опция метода на
% Монте Карло, като се дисконтира очакваната цена
Euro_Call_Value = exp(-rate*T)*(sum/M)
% Генериране на фигура с ценовите пътеки SPaths
plot(t, S)
%Етикети за заглавие и оси на генерираната фигура
title('Симулация на ценови пътеки с процес на Леви в модела на
Мертон ', 'FontSize',12)
xlabel('Време t до падежа T, T=2')
ylabel('30 ценови пътеки генерирани от S_{0}=35')

```



Фиг. 2. Симулация на ценови пътеки в модела на Мертон

ИЗВОДИ

При генерирането на ценови пътеки в модела на Мертон трябва да бъдат симулирани два случайни процеса – един, който описва скоковете на цената на дадения актив (Поасонов) и дифузионен (Брауново лутане). Важно за симулацията на ценови пътеки в модела на Мертон е, че двата случайни процеса са независими един от друг [1].

Прилагането на метода Монте Карло не изисква задълбочени математически познания в областта на интегралното и диференциалното смятане за разлика от други количествени методи като квадратурния метод и схемите на крайни разлики. Структурата на метода Монте Карло изисква познания от линейната алгебра като умножение на числа с матрици, събиране и умножение на матрици, поради това той лесно може да бъде внедрен чрез системата MATLAB (MatrixLaboratory) [1].

REFERENCES

- [1] Milev, M. (2012). *Application of Matlab for Modeling and Evaluation of Financial Derivatives*. Econometrics, July 2012, Sofia.
- [2] Dimov, I. (2012). *Advanced Monte Carlo Methods – Computational Challenges*, ИКТ, Bulgarian Academy of Sciences.
- [3] URL: https://en.wikipedia.org/wiki/Monte_Carlo_method (Accessed on 28.04.2018).

OPTION PRICING USING MONTE CARLO METHODS⁴

Boyan Boyanov – Student

Financial Mathematics

University of Ruse “Angel Kanchev”

E-mail: bboyanov@abv.bg

Abstract: Option Pricing Using Monte Carlo Methods: Choosing the right derivatives to make a portfolio is a complicated task as is choosing the right mathematical model for pricing them. In this article we will examine the advantages and disadvantages of the different models for options pricing, which will help us to choose the best one. We will make a comparison of the Black-Scholes' and Merton's models using the Monte Carlo method. Our results suggest that Merton's model is better as the possibility of appearance of jumps in the price paths. It is important to make difference between the models because making a mistake in these calculations would cost big losses of the portfolio owner.

Key words: Monte Carlo Methods, option, pricing, Black-Scholes, Merton

ВЪВЕДЕНИЕ

Опциите са финансови деривати, които дават правото, но не и задължението, на собственика си, да купи или продаде даден актив на предварително уговорена цена и дата. Купувачът плаща определена премия за това си право. Те биват два вида: кол и пут опции. Кол опциите дават правото на собственика да купи даден актив, а пут опциите – да продаде. Чрез опциите могат да се търгуват активи като : акции, облигации, фючърси, форуърди, индекси, суровини и валути. Има няколко различни типа опции по отношение на датата на изпълнение: американски, европейски, бермудски, наричани още ванилови опции и азиатски, бариерни, лукбак, наричани още екзотични опции.

Много инвеститори търгуват с бариерни опции, за да ограничат риска от големи загуби. На опцията могат да бъдат поставени горна (U) и долна (L) бариера, за да се избегнат големи загуби. Опцията губи валидност, ако преди падежа цената на дадения актив е попаднала извън бариерния коридор (L,U). Опциите с бариери се оценяват с методи Монте Карло.

Методите Монте Карло имат много приложения в различни области. Например, във физиката са в основата на съвременното прогнозиране на времето и проектиране на аеродинамични форми. В телекомуникациите, при планирането на безжична мрежа, проектът трябва да се докаже, при голямо разнообразие от сценарии, които зависят главно от броя на потребителите, техните места и услугите, които те използват. Методите Монте Карло обикновено се използват за генериране на тези потребители.[3]

В статията е пресметнат премиума на европейска кол опция с бариери. Експериментът е проведен с методи Монте Карло като са приложени моделите на Блек-Шолс и Мертон. Наблюдава се как интензитета на скоковете и бариерната честота влияят върху премиума на опцията.

ИЗЛОЖЕНИЕ

В случай на дискретно наблюдавана нокаут кол опция с две бариери, функцията на печалбата има вида на ванилова Европейска кол опция, т.е. $\max(S-K, 0)$, но опцията изтича без стойност, ако преди падежа y , цената на разглеждания актив S е попаднала извън бариерния коридор $[L, U]$, в който и да е момент t преди падежа T , т.е. за $t \in [0, T)$ имаме $V(S, t) = 0$, ако $S \notin (L, U)$ и при $t = T$:

⁴ Докладът е представен на студентската научна сесия на 23. 05. 2018 г. в секция „Финансова математика“ с оригинално заглавие на български език: ОЦЕНЯВАНЕ НА ОПЦИИ С МЕТОДИ МОНТЕ КАРЛО.

$$V(S,T) = \begin{cases} 0, \text{ ако } S \leq L, \\ \max(S - K, 0), \text{ ако } L < S < U, \\ 0, \text{ ако } S \geq U \end{cases}$$

т.е. функцията на печалбата е всъщност условието $V(S,T)$ при $t=T$. Пресмятането на премиума на дискретни нокаут кол опции с две бариери имаме

$$e^{-rt} E[\max(S_t - K, 0) \cdot 1_{\{A_1\}} 1_{\{A_2\}} \dots 1_{\{A_n\}}], A_i = \{S_t \in (L, U)\}.$$

В случай че една от бариерите е достигната и опцията е анулирана, притежателят може да бъде компенсирал с допълнително заплащане R , което е предварително определено. В този случай опцията губи валидност, функцията на печалбата е нула, но премиумът на опцията не е нула, а се определя по формулата

$$V(L, t_i) = e^{-rt_i} R_L, V(U, t_i) = e^{-rt_i} R_U,$$

където t_i е момент, когато бариерите L и U се прилагат и цената на разглеждания актив S_{t_i} е достигнала една от тях (дискретните моменти t_i са фиксирани).

В случай, че не се изплащат компенсации при анулиране на опции с бариери, следната зависимост между премиумите на опции, съответно с постоянни бариери, дискретни бариери и обикновени ванилови опции без бариери е изпълнена:

$$V_{\text{нокаут_опция_с_постоянни_бариери}} \leq V_{\text{дискретна_нокаут_опция}} \leq V_{\text{ванилова_опция_без_бариери}}.$$

В модела на Блек-Шолс равенство може да се достигне, когато бариерите са твърде отдалечени от страйковата цена, например горната бариера е фиксирана да бъде два пъти страйка.

Нека пресметнем премиума на Европейска дискретна нокаут кол опция с две бариери L и U , дефинирана с функция на печалба $\max(S-K, 0)$ в датата на падежа T , страйкова цена $K=40$, параметър на волатилност 10% годишно, падеж една година, 8% безрисков лихвен процент, образуван на годишна база, долна бариера $L=35$, горна бариера $U=50$ и начална цена на разглеждания актив $S_0=40$.

Ще изследваме премиума на нокаут опции с бариери в модела на Мертон за *различни стойности на параметъра λ* , описващ интензитета на скоковете за единица време. За яснота бариерната честота означаваме с m .

Ще проведем експеримент с моделите на Блек-Шолс и Мертон като приложим метода Монте Карло чрез 10^6 ценови пътеки, при петкратно прилагане на бариери, седмично и дневно наблюдение и различни стойности на интензитета. Ще наблюдаваме влиянието на

- интензитета на скоковете;
- бариерната честота;
- бариерната честота и интензитета на скоковете върху премиума на опцията.

Резултатите са представени в таблица 1, където $L=35$ и $U=50$, $S_0=40$, $K=40$, $T=1$, $\sigma=0,01$ и $r=0,08$. При опции с падеж 1 година параметърът m има стойности $m=50$ и $m=250$, съответно при седмично и дневно наблюдение на опцията. При петкратно прилагане на бариери за параметъра m имаме $m=5$.

В модела на Мертон анализът на резултатите показва, че с увеличаване на стойността на λ , премиумът на нокаут кол опция с дискретни бариери намалява от 2,4142 до 0,9643, когато λ се увеличава от $\lambda=5$ до $\lambda=25$. Това може лесно да бъде обяснено като се вземе предвид право пропорционалната зависимост между възможността да се пресече една от бариерите и стойността на параметъра λ , описващ честотата на скоковете.

Тази закономерност се потвърждава и от процента, който показва броя на ценовите пътеки пресекли една от бариерите, към общия брой симулирани ценови пътеки, т.е. стойността на $100 \cdot (N_{\text{Crossed}} / N_{\text{Repl}})$ в програма 2. Този процент за $\lambda=5$ е 20,96%, докато за

$\lambda = 25$ се е увеличил до 38%. Ще отбележим, че този извод за увеличението на разглеждания процент от пресекли ценови пътеки е валиден и за по-малки стойности на параметъра волатилност σ , т.е. при $\sigma < 0,1$. Този феномен може да се обясни със значимостта на параметъра на интензитет λ в модела на Мертон, който описва 'активността' на ценовите пътеки.

Таблица 1. Премиум на Европейска дискретна нокаут кол опция с две бариери.

Наблюдавана честота на бариерите	Модел на Блек-Шолс без скокове	Модел на Мертон със скокове при параметър на интензитет λ	
m	премиум на опцията (стандартна грешка)	λ	премиум на опцията (стандартна грешка)
5	2,6899 (0,0052)	5	2,4142 (0,0036)
5	2,6899 (0,0052)	10	1,9069 (0,0040)
5	2,6899 (0,0052)	25	0,9643 (0,0038)
50	2,5368 (0,0050)	5	2,4075 (0,0036)
50	2,5368 (0,0050)	10	1,8824 (0,0040)
50	2,5368 (0,0050)	25	0,9069 (0,0036)
250	2,4746 (0,0050)	5	2,4019 (0,0036)
250	2,4746 (0,0050)	10	1,8764 (0,0020)
250	2,4746 (0,0050)	25	0,8982 (0,0036)

В първоначалния модел на Блек-Шолс отсъства параметър, който характеризира евентуална поява на скокове в ценовия процес S_t . Можем условно да считаме, че параметърът λ е винаги 0 в модела на Блек-Шолс, т.е. $\lambda=0$, което обяснява наблюдаваните малки изменения на стойностите на симулираните ценови пътеки в последователни времеви моменти [1]. Не случайно, при $\lambda=0$ частното диференциално уравнение описващо премиума на опции в модела на Мертон, се трансформира от

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + (r - d - \lambda \hat{k})S \frac{\partial V}{\partial S} - rV + \lambda E_k[V(S(1+k), t) - V(S, t)] = 0$$

в известното фундаментално уравнение на Блек-Шолс:

$$\frac{\partial V}{\partial t} + rS \frac{\partial V}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} - rV = 0$$

Изборът на математически модел за оценяване на опции е от голямо значение за притежателя на опция с бариери в случаите, когато опцията се анулира при достигане на бариерите и той очаква компенсация. Избраният модел и даденият финансов дериват определят до голяма степен и използването на подходящ числен метод. Разбира се това зависи от поставените цели като достигане на точност и изчислително време или комбинация от тези два фактора [2].

По отношение на влиянието на бариерната честота върху премиума на нокаут опци, можем да разграничим първоначалния модел на Блек-Шолс с едно от неговите разширения модела на Мертон, в който ценовите пътеки се характеризират с възможността от поява на скокове. В модела на Мертон за фиксиран параметър $\lambda=5$, при честоти 5, 50 и 250, премиумът на нокаут опция е съответно 2,4142; 2,4075 и 2,4019. Същият извод е валиден и когато фиксираме $\lambda=10$, където премиумите са съответно 1,8824 и 1,8764 и при $\lambda=25$, съответно 0,9069 и 0,8982, като имаме предвид честотата само при седмично и дневно наблюдение.

Обикновените ванилови опции нямат бариери, т.е. честотата е $m=0$, но премиумът им е винаги по-голям от опциите с бариери. В таблицата със стрелка сме отбелязали посоката на

растене на премиума в реда. При нокин опциите, вероятността да се активира опцията е право пропорционална на честотата на бариерите, валидна е зависимостта:

$$V_{\text{ванилова опция без бариери}} \geq V_{\text{нокин опция с постоянни бариери}} \geq V_{\text{дискретна нокин опция}},$$

Таблица 2. Изследване на премиума на нокаут опции с две бариери $L=95$ и $U=140$ в зависимост от бариерната честота m при падеж $T=0,5$ и параметри $S_0=100$, $K=100$, където $\sigma=0,2$ и $r=0,1$.

Нокаут опция	Постоянни бариери	Дискретни бариери	Дискретни бариери	Дискретни бариери	Ванилова опция
честота	$m \geq 10^6$	$m=250$	$m=50$	$m=5$	$m=0$
премиум	→	→	→	→	→

Премиумът на нокин опцията намалява когато честотата намалява, т.е. точно обратно на нокаут опцията. Ще отбележим, че дори и при много големи стойности на m , като например $m=10^6$, премиумът на опциите с дискретни бариери се различава значително от този на опции с постоянни бариери. Не е целесъобразно математическо изследване на дискретния случай на наблюдение чрез непрекъснатост за достатъчно големи стойности на m , защото практически $m \leq 250$ [1].

Програма 1

```
% AssetJumpC.m
function SPaths = AssetJumpC(S0, K, r, T, sigma, lambda, Nsteps, NRepl)
Gamma=0; Delta=0.3; Gamma1=Gamma-0.5*Delta^2;
kappa = exp(Gamma) - 1; dt = T/Nsteps;
for i = 1 : NRepl
    SPaths(1,i) = S0; % начална стойност на актива
    for j = 2:Nsteps+1
        % Генериране на случайните числа чрез Нормално разпределение
        deviate = normrnd(0,1);
        % Генериране на случайни числа чрез Поасоново разпределение
        temp = poissrnd(lambda*dt); TJS = 0;
        for k = 1:temp
            JS = normrnd(Gamma1,Delta); TJS = TJS + JS;
        end
        SPaths(j,i) = SPaths(j-1,i)*
            exp((r-lambda*kappa - 0.5*sigma^2)*dt +
            sigma*sqrt(dt)*deviate + TJS);
    end
end
end
```

Програма 2

% Метод Монте Карло за оценяване на дискретни нокаут опции с две бариери в модела на Мертон

% MCEJump.m

function[D,CI,NCrossed] =

MCEJump(S0,K,r,T,sigma,lambda,L,U,NSteps,NRepl)

% Генериране на ценовите пътеки

Payoff=zeros(NRepl,1);

NCrossed =0;

for i=1:NRepl

 Path = AssetJumpC(S0,K,T,r,sigma,lambda,NSteps,1);

 crossed = any(Path <=L | Path >=U);

 if crossed ==0

 Payoff(i)=max(0,Path(NSteps+1)-K);

 % В случай на пут опция, функцията на печалба е:

 % Payoff(i)=max(0,K-Path(NSteps+1));

 else

 Payoff(i)=0;

 NCrossed=NCrossed+1;

 end

end

% Определяне премиума на опцията и съответния 95%

% доверителен интервал за средната цена

[D,aux,CI]=normfit((exp(-r*T))*Payoff);

% Процент ценовите пътеки (спрямо общия брой

% симулирани), които са пресекли бариерите е:

100*(NCrossed/NRepl)

ЗАКЛЮЧЕНИЕ

В статията става ясно, че моделът на Мертон прави по-точна оценка на бариерните опции, защото включва в себе си параметър, който описва скоковете на ценовите пътеки.

REFERENCES

[1] N. Velinova, *Pricing the risk of a portfolio of derivatives using Monte Carlo Methods*, CFC, Bulletin 1/2018.

[2] M. Milev, *Matlab Application for Modeling and Pricing of Financial Derivatives*, "Evdemoniya Production Ltd", Sofia 2012.

[3]

https://bg.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%9C%D0%BE%D0%BD%D1%82%D0%B5_%D0%9A%D0%B0%D1%80%D0%BB%D0%BE

MODEL OF LIQUIDITY ANALYSIS WITH EXCEL⁵

Byulent Idirizov – Student

Mathematical Modelling in Finances, Insurance and Social Work,
Faculty of Natural Sciences and Education,
University of Ruse “Angel Kanchev”
E-mail: byulent.idirizov@bulmarket.bg

Assoc. Prof. Lyubomir Lyubenov, PhD

Department of Economics,
Faculty of Business and Management,
University of Ruse “Angel Kanchev”
E-mail: Llyubenov@uni-ruse.bg

Abstract: *Mathematical Model for Financial Accounting Analysis of an Enterprise's Liquidity Created in Excel.* A mathematical model conducting automated liquidity analysis of an enterprise after entering data from the balance sheet. It presents not only the actual state of the liquidity ratios, but also how they can be improved. Improving liquidity ratios depends on the change in short-term liabilities and short-term assets. The model takes into account the foreseeable change in short-term liabilities and determines how the individual elements of short-term assets should be changed.

Keywords: Financial condition; Insolvency; Liquidity; Mathematical model; Excel; Accounting.

ВЪВЕДЕНИЕ

Ликвидността е способността на предприятието да превръща краткотрайните си активи в парични средства, за да покрива краткосрочните си задължения в кратък срок и при минимум загуби. Съществуват няколко показателя (коефициенти) за ликвидност. Представен е модел, изработен в Excel за анализ на коефициентите на ликвидност на търговско предприятие, след въвеждане на данни за стойността на краткосрочните пасиви и за стойностите на видовете краткотрайни активи. Освен анализ, моделът също представя и решение на проблема за подобряване на ликвидността при предварително зададени целеви оптимални коефициенти.

ИЗЛОЖЕНИЕ

Капиталът вложен в краткотрайните активи има за цел да осъществява непрекъснатостта на производствения процес, платежоспособност и ликвидност на предприятието. Рационалното използване на оборотните средства води до намаляване себестойността на продукцията, увеличаване на производителността, подобряване на качеството, икономично изразходване на материалните запаси и др., което увеличава и рентабилността. [1]

Ликвидността се дефинира като количествена характеристика на способността на дадено предприятие да изплаща краткосрочните си задължения с наличните краткотрайни активи, което е равносилно на способността на активите на предприятието да се обръщат в парични средства, които са най-ликвидни. Казано по друг начин, ликвидността е показател за финансовото равновесие между краткотрайните активите и краткосрочните задължения. [4] Важно е да има съответствие между сумата на предстоящите плащания и сумата на разполагаемите за тях ликвидни активи, тъй като плащанията са различни във времето и за целта се изчисляват различни показатели за ликвидност, като се посочват и границите за тяхното изменение. [2]

⁵ Докладът е представен на студентската научна сесия на 23. 05. 2018 г. в секция „Финансова математика“ с оригинално заглавие на български език: МОДЕЛ ЗА АНАЛИЗ НА ЛИКВИДНОСТТА С EXCEL.

Ликвидността на предприятието бива статична и динамична. В модела е наблегнато на анализа на статичната ликвидност, която е способността на предприятието да погасява краткосрочните си осчетоводени задължения. То се установява с помощта на счетоводния баланс и справката за вземанията, задълженията и провизиите (Елемент № 7.3 на приложението към финансовия отчет). В тях се съдържа информация, позволяваща изчисляване на показателите за ликвидност. Величините на тези показатели трябва да бъдат оповестени в справката за оповестяване на счетоводната политика, тъй като съществува такова изискване в СС 1 Представяне на финансови отчети (т. 19.3.5). [3] В т. 16 от известните ни Общи положения за изготвяне и представяне на финансови отчети, които са одобрени от Съвета по международни счетоводни стандарти, се казва, че “под ликвидност се разбира наличността на парични средства в близко бъдеще при отчитане на финансовите задължения за същия период”. Всъщност написаното от нас относно ликвидността е именно в духа на това определение за ликвидността на фирмата. Показателите за статична ликвидност са: [3] Коефициент на обща ликвидност (Кол), коефициент на бърза ликвидност (Кбл), коефициент на незабавна ликвидност (Кнл) и коефициент на абсолютна ликвидност (Кал).

Първо ще разгледаме коефициента на абсолютна ликвидност (Кал). Той е равен на съпоставянето на паричните средства към краткосрочните пасиви или:

$$\text{Кал} = \frac{\text{Парични средства}}{\text{Краткосрочни пасиви}}$$

Както се забелязва, при изчисляването му се използват най-бързоликвидни активи, или казано по друг начин, елиминирани са всички активи, които са по-бавно ликвидни от паричните средства. Трябва да се отбележи също, че паричните средства могат да варират в различни валути и поради което, в зависимост от случая превръщането им към валутата, с която се търгува в съответната държава, понякога може да отнеме до няколко дена.

Препоръчителните граници за коефициента на абсолютна ликвидност са от 0.1 до 0.2. Следователно всяко съотношение на паричните средства към краткосрочните пасиви, което е по малко 0.1 означава, че предприятието не разполага с нужните налични парични средства за покриване на краткосрочните си задължения. Когато Кал е над 0.2, тогава можем да направим извода, че предприятието има налични парични средства, които не оползотворява, било под формата на краткосрочни инвестиции, покупка на нови материални запаси или под някоя друга форма, с цел използването на излишните парични средства в дейността на предприятието. В зависимост от отрасловата и браншовата специфика, от финансовия и производствения цикъл на предприятието оптималния коефициент на абсолютна ликвидност може да бъде в граници от 0.1 до 0.2. В зависимост от актуалния коефициент на абсолютна ликвидност и оптималния коефициент за даденото предприятие, може да бъде изчислено как да бъдат регулирани и на каква стойност да бъдат равни паричните средства.

Трябва да се знае каква е предвидената промяна на краткосрочните пасиви. Ако с k отбележим предвидената промяна на краткосрочните пасиви (КП), то тя може да бъде положителна, отрицателна или равна на нула (0). С КП_2 отбелязваме краткосрочните пасиви, след извършване на предвидената промяна на тяхната стойност $\text{КП}_2 = \text{КП} + k$. Отбелязваме също с p сумата, с която трябва да бъде променена стойността на паричните средства за да бъде коефициента на абсолютна ликвидност в изискуемите граници. Ако Кал_2 е оптималния коефициент за даденото предприятие, който е поставен за цел, тогава:

$$\text{Кал}_2 = \frac{\text{ПС} + p}{\text{КП} + k}, \quad \text{където } \text{Кал}_2 \in (0,1; 0,2).$$

От тази формула може да бъде изведена зависимост за p :

$$p = \text{Кол}_2 * (\text{КП} + k) - \text{ПС}.$$

Формулата за промяната на ПС, представена като проценти от ПС изглежда по следния начин:

$$p\% = \frac{Кал_2}{ПС/(КП + k)} - 1.$$

При изчисленията, в никакъв случай не трябва да се пренебрегва факта, че при промяната на стойностите на отделните елементи на краткотрайните активи, се променя и цялата стойност на краткотрайните активи, също че в зависимост от това как се планира да бъдат използвани паричните средства и другите краткотрайни активи, може да бъде изменена и стойността на краткосрочните пасиви.

На следващо място ще разгледаме коефициента на незабавна ликвидност (Кнл), който е равен на съпоставянето на паричните средства и краткосрочните инвестиции (КИ) към краткосрочните пасиви или:

$$Кнл = \frac{Парични\ средства + Краткосрочни\ инвестиции}{Краткосрочни\ пасиви}$$

При изчисляване на този коефициент се изключват краткосрочните вземания и материалните запаси. Коефициента на незабавна ликвидност (Кнл) показва способността на предприятието да обръща активите в пари, в срок от няколко седмици.[2] Той е особено необходим за изчисляване при предприятия, имащи трудно инкасиращи се краткосрочни вземания. В зависимост от отрасловата и браншова специфика, от финансовия и производствения цикъл на предприятието, оптималния коефициент на незабавна ликвидност може да бъде в граници от 0.2 до 0.3. В зависимост от актуалния коефициент на незабавна ликвидност, от оптималния коефициент за даденото предприятие, и от изчислената преди това (p) – сумата, с която трябва да бъде променена стойността на паричните средства, може да бъде изчислено как да бъдат регулирани и каква трябва да бъде стойността на краткосрочните инвестиции. Отбелязваме също с n сумата, с която трябва да бъде променена стойността на краткосрочните инвестиции, за да бъде коефициента на незабавна ликвидност в изискуемите граници. Ако $Кнл_2$ е оптималния коефициент на незабавна ликвидност за даденото предприятие, който трябва да бъде постигнат, тогава:

$$Кнл_2 = \frac{ПС + p + КИ + n}{КП + k}, \text{ където } Кнл_2 \in (0,2; 0,3)$$

От тази формула може да бъде изведена зависимост за n :

$$n = Кнл_2 * (КП + k) - ПС - p - КИ,$$

Формулата за промяната на КИ, представена като в проценти от КИ изглежда по следния начин:

$$n\% = \frac{n}{КИ} = \frac{Кнл_2 * (КП + k) - ПС - p - КИ}{КИ},$$

Следващият коефициент, който ще бъде разгледан е коефициента на бърза ликвидност (Кбл). Той е равен на съпоставянето на краткотрайните активи без материалните запаси (МЗ) към краткосрочните пасиви или:

$$Кбл = \frac{Парични\ средства + Краткосрочни\ инвестиции + Краткосрочни\ Вземания}{Краткосрочни\ пасиви}$$

Коефициентът на бърза ликвидност показва способността на предприятието да обръща наличните си активи в пари в рамките на няколко месеца. Този коефициент в по-голям степен

дава представа за реалните парични средства, с които предприятието разполага, и които по-бързо могат да бъдат изплатени на кредиторите. [3] Трябва да се отбележи, че стойността на краткосрочните вземания, трябва да бъде контролиран в размер по-малък от 40% от краткосрочните пасиви ($KB < 0.4KP$), и по време стойността на текущите краткосрочни вземания трябва да бъде по малък от размера на текущите краткосрочни пасиви ($Tkv < Tkp$). Този контрол позволява не само по-добра ликвидност, но и по-добра обращаемост, следователно и рентабилност. В зависимост от отрасловата и браншова специфика, от финансовия и производствения цикъл на предприятието, оптималния коефициент на бърза ликвидност може да бъде в граници от 0.5 до 0.6. В зависимост от актуалния и оптималния коефициент на бърза ликвидност за даденото предприятие, от изчислената преди това (p) и (n) - сумите, с които трябва да бъдат променени стойностите на паричните средства и краткосрочните инвестиции, може да се изчисли как да бъдат регулирани и каква трябва да бъде стойността на краткосрочните вземания.

Ако с l отбележим сумата, с която трябва да бъде променена стойността на краткосрочните вземания (KB) за да бъде коефициента на бърза ликвидност в изискуемите граници. Когато Kbl_2 е оптималния коефициент на бърза ликвидност за даденото предприятие, който трябва да бъде постигнат, тогава:

$$Kbl_2 = \frac{ПС + p + KI + n + KB + l}{KP + k}, \text{ където } Kbl_2 \in (0.5; 0.6)$$

От тази формула може да бъде изведена зависимост за l :

$$l = Kbl_2 * (KP + k) - ПС - p - KI - n - KB,$$

Формулата за промяната на KB , представена като проценти от KB изглежда по следния начин:

$$l\% = \frac{l}{KB} = \frac{Kbl_2 * (KP + k) - ПС - p - KI - n - KB}{KB},$$

Последният коефициент, който ще бъде разгледан е коефициента на обща ликвидност (Кол). Той показва способността на предприятието да обръща наличните си активи в пари в рамките на една година [3]. Кол е равен на съпоставянето на краткотрайните активи (КА) към краткосрочни пасиви или:

$$Кол = \frac{ПС + KI + KB + M3}{KP} = \frac{КА}{KP}$$

В силна зависимост от отрасловата и браншовата специфика, от финансовия и производствения цикъл на предприятието, оптималния коефициент на обща ликвидност може да бъде в граници от 1 до 2. Ако Кол е по-малък от 1, тогава може да се направи извода, че предприятието не е ликвидно, има голяма стойност от натрупани задължения и недостатъчна стойност на краткотрайните активи с които да може да ги покрие, което може да се дължи на небалансирано и неправилно управление на активите на предприятието. Когато стойността на $Кол > 2$, се счита че предприятието е свръх ликвидно, има излишък от средства, които не успява да оползотвори в своя бизнес процес. Нерационалното използване на паричните средства, забавеното разплащане с клиенти и забавената обращаемост на материалните запаси, могат също да бъдат причини за прекалено високата стойност на Кол. Трябва да се има предвид, че една значителна част от оборотния капитал е ангажирана в материалните запаси, които не винаги могат да се отнесат в категорията бързо ликвидни, което е недостатък на Кол [3]. Поради това, се препоръчва, при анализа да се използват и другите показатели, така че да

бъде допълнена информацията за равнището на ликвидността и да бъде повишена нейната достоверност.

За да бъде оптимизирана работата на предприятието може да се пресметне как и в каква степен да бъдат регулирани краткотрайните активи. Намаляването и увеличаването на КА водят съответно до намаляване и увеличаване на Кол. Понеже КА включват ПС, КИ, КВ и МЗ, а ние в горните стъпки сме изчислили как да бъдат регулирани ПС, КИ и КВ, тук ще наблегнем на регулацията на МЗ. Следователно, в зависимост от актуалния и оптималния коефициент на обща ликвидност за даденото предприятие, и от изчислените преди това (p), (n) и (l), може да бъде изчислено каква трябва да бъде стойността на материалните запаси.

Ако m е сумата, с която трябва да бъде променена стойността на материалните запаси (МЗ) за да бъде коефициента на обща ликвидност в изискуемите граници. Ако Кол₂ е оптималния коефициент на обща ликвидност за даденото предприятие, който трябва да бъде постигнат, тогава:

$$\text{Кол}_2 = \frac{\text{ПС} + p + \text{КИ} + n + \text{КВ} + l + \text{МЗ} + m}{\text{КП} + k} = \frac{\text{КА} + p + n + l + m}{\text{КП} + k},$$

където Кол₂ ∈ (1; 2)

От тук може да бъде изведена зависимост за m :

$$m = \text{Кол}_2 * (\text{КП} + k) - \text{ПС} - p - \text{КИ} - n - \text{КВ} - l - \text{МЗ},$$

или

$$m = \text{Кол}_2 * (\text{КП} + k) - \text{КА} - p - n - l,$$

Формулата за промяната на МЗ, представена като проценти от МЗ изглежда по следния начин:

$$m\% = \frac{m}{\text{МЗ}} = \frac{\text{Кол}_2 * (\text{КП} + k) - \text{КА} - p - n - l}{\text{МЗ}},$$

Пример: Определете ликвидността и направете модел за регулиране на КА на предприятие “Х” за период след една година, при условие, че в рамките на една година е планирано увеличаване на КП с 10%. Ако текущата стойност на МЗ=60 000 лв., КВ=25 500 лв., КИ=1 000 лв., ПС=24 000 лв. и КП=65 000 лв. Според експертно мнение, коефициентите на ликвидност за предприятието трябва да бъдат съответно приблизително равни на Кол=0.2, Колл=0.25, Колл=0.6, Колл=1.2.

	Текуща година	Коефициенти на ликвидност	Текуща година	Целеви стойности
Краткотрайни Активи	110 500.00	Коефициент на абсолютна ликвидност	0.3692	0.20
Парични Средства	24 000.00	Коефициент на незабавна ликвидност	0.3846	0.25
Краткосрочни Инвестиции	1 000.00	Коефициент на бърза ликвидност	0.7769	0.60
Краткосрочни Вземания	25 500.00	Коефициент на обща ликвидност	1.7000	1.20
Материални Запаси	60 000.00			
Краткосрочни Пасиви	65 000.00			
Планирана промяна на Краткосрочни Пасиви в лв.	6 500			
Планирана промяна на Краткосрочни Пасиви в %	10.00%			
Краткосрочни Пасиви в лв. след планираната промяна	71 500.00			

Според показателите от текущата година

Паричните Средства	трябва да бъдат намалени с	
	40.42%	9 700 лв.
Краткосрочните Инвестиции	трябва да бъдат увеличени с	
	257.50%	2 575 лв.
Краткосрочните Вземания	трябва да бъдат намалени с	
	1.86%	475 лв.
Материалните Запаси	трябва да бъдат намалени с	
	28.50%	17 100 лв.

Решение: След въвеждане на данни за стойността на краткотрайните активи и краткосрочните пасиви на предприятието, и за желаните нива на коефициентите за ликвидност в модела изготвен в Excel, моделът ни извежда резултат. Може да бъде направен извода, че в рамките на периода, през който е планирано увеличаването на краткосрочните пасиви с 10%, трябва:

- Паричните средства да бъдат намалени с 9 700 лв., което означава, че трябва да бъдат използвани над 40% от тях и да се оставят налични около 14 300 лв.
- Краткосрочните инвестиции трябва да бъдат увеличени с 2 575 лв. или над два и половина пъти.
- Стойността на краткосрочните вземания трябва да бъде намалена с 475 лева, или с близо 2%.
- Предприятието трябва да намали материалните си запаси с над 28.5%, или със 17 100 лв.

ЗАКЛЮЧЕНИЕ

Предложен е модел на Excel, извеждащ математически отговор на това как трябва да бъдат регулирани отделните елементи на краткотрайните активи, за достигане на оптимални нива на коефициентите за ликвидност. Чрез него се автоматизират изчисленията, подобрява се контрола върху ликвидността и обращаемостта. Приложението на модела в практиката може да спести значителни средства и време на предприятията.

REFERENCES

- [1] Любенов, Л., *Бизнес Икономика*, Русе, 2017.

- [2] Любенова, А., *Устойчивост на финансовото състояние на земеделските стопанства*, Русе, 2015.
- [3] Petrov P. P., *Cycle of working capital and liquidity – additional analyzes*, 2000
- [4] <http://schetovodstvo.dokumentite.com/art/analiz-na-likvidnostta-10/82781>

NUMERICAL COMPUTATION OF THE IMPLIED VOLATILITY SURFACE FROM REAL-TIME MARKET DATA USING LELAND MODEL⁶

Slavi Georgiev – Student

Mathematical Modelling in Finance, Insurance and Social Work,
Faculty of Natural Sciences and Education,
University of Ruse “Angel Kanchev”, Ruse, Bulgaria
Tel.: +359 889 021 262
E-mail: aslv@abv.bg

Assoc. Prof. Beáta Stehlíková, PhD

Department of Applied Mathematics and Statistics,
Faculty of Mathematics, Physics and Informatics,
Comenius University, Bratislava, Slovakia
E-mail: stehlikova@pc2.iam.fmph.uniba.sk

Abstract: In this paper the Leland model for option pricing is considered, which is an effective and successful extension to the widely spread Black–Scholes framework, making use of the latter with a volatility adjustment. It is famous for taking the transaction costs into account. Also, the iterative Newton–Raphson method for finding the implied volatility is employed due to its simplicity and high convergence rate. The implied volatility is of particular importance since it is the main ingredient in the calculation of the option price. Furthermore, the algorithm for deriving the implied volatility values and surface is presented and discussed. Numerical experiments with real market data are performed which support the conclusions and explain the results.

Keywords: Implied Volatility Surface, Leland Model, Black–Scholes Framework, Newton Root-Finding Method
JEL Codes: C02, G12, G13

INTRODUCTION

For all assets and derivatives including options, one of the most fundamental problems is deriving their “fair” price. The market is arguably the mechanism that makes the prices, so we have to model the market in order to find a reasonable option price. In this paper we adopted the model [5] named after Hayne Leland which could be viewed as an extension and improvement to still ubiquitously used Black–Scholes framework.

Let us briefly introduce the basic assumptions which the Black–Scholes model [1] is based on. To begin with, we assume the existence of at least one risky asset, called underlying asset, and a riskless asset, e. g. zero coupon bond. Moreover, the riskless interest rate is a constant; the price of the underlying asset follows a geometric Brownian motion. The following assumptions are posed on the market: an arbitrage is not allowed, thus all market participants have equal, immediate and unlimited information; borrowing and lending money of all quantity is allowed at the riskless rate as well as longing or shorting the underlying asset; individual trade doesn’t affect the market price [2].

In the sake of comprehensiveness and punctuality we have to mention that the Black–Scholes model assumes the market is frictionless, i. e. there are no transaction costs. This is the simplification of the reality which the Leland model doesn’t require as discussed later. So, the famous PDE of Black and Scholes is stated as:

$$\frac{\partial V}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} - rV = 0, \quad (1)$$

⁶ Докладът е представен на студентската научна сесия на 23. 05. 2018 г. в секция „Финансова математика“ с оригинално заглавие на български език: ЧИСЛЕНО ПРЕСМЯТАНЕ НА ПОВЪРХНИНАТА НА КОСВЕНАТА ВОЛАТИЛНОСТ ОТ ПАЗАРНИ ДАННИ В РЕАЛНО ВРЕМЕ, ИЗПОЛЗВАЙКИ МОДЕЛА НА ЛЕЛАНД.

with the following well-known analytical solutions for European call and put options:

$$V_{call}^{BS} = SN(d_1^{BS}) - Ke^{-r(T-t)}N(d_2^{BS}), \quad V_{put}^{BS} = -SN(-d_1^{BS}) + Ke^{-r(T-t)}N(-d_2^{BS}),$$

where

$$N(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{s^2}{2}} ds$$

is the cumulative distribution function of the standard normal distribution and

$$d_1^{BS} = \frac{\ln \frac{S}{K} + \left(r + \frac{\sigma^2}{2}\right)(T-t)}{\sigma\sqrt{T-t}}, \quad d_2^{BS} = \frac{\ln \frac{S}{K} + \left(r - \frac{\sigma^2}{2}\right)(T-t)}{\sigma\sqrt{T-t}} = d_1^{BS} - \sigma\sqrt{T-t}.$$

In the original temporal variable the PDE is backward with the following boundary and terminal conditions (for a call): $V(S, T) = (S - K)^+$, $V(0, t) = 0$, $V(S, t) = S, S \rightarrow \infty$. It can also be shown that $V(S, t) \sim S - Ke^{-r(T-t)}, S \rightarrow \infty$. The boundary conditions for an European put option are derived in the same manner.

As stated above, the absence of transaction costs is one of the main limitations of the Black–Scholes framework. There are many ways for this feature to be overcome. One of the most widely used approaches is the one which was exploited by Leland. It is successful because it is independent from the individual investor's risk preference and has a closed-form solution, while the others are time consuming to compute.

Definitely, estimating transaction costs is an important topic from both theoretical and empirical points of view. They consist of bid-ask spreads, commissions, brokerage and other costs related to security trading [6].

Now we will briefly explain the idea of Leland. He developed a hedging strategy in which the price of an European option is given by the Black–Scholes model with an adjusted volatility. The latter depends on the proportional transaction costs rate, the volatility of the underlying asset and the rebalancing interval of the replicating portfolio [6]. He claimed that the hedging error could be arbitrary small if the length of the rebalancing frequency tends to zero and if one uses the Black–Scholes delta-hedging strategy with the adjusted volatility (3).

Let us denote

$$S = \frac{S_{ask} + S_{bid}}{2}, \quad c = \frac{S_{ask} - S_{bid}}{S},$$

and we hedge discretely on intervals of minimum Δt time (years). It is impossible for us to hedge continuously due to the presence of transaction costs [8]. Now, using the replication portfolio arguments and the Leland's propositions we arrive at the final form of the PDE:

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} \left[1 - \frac{c}{\sigma\sqrt{\Delta t}} \sqrt{\frac{2}{\pi}} \operatorname{sgn} \left(\frac{\partial^2 V}{\partial S^2} \right) \right] + rS \frac{\partial V}{\partial S} - rV = 0. \quad (2)$$

It is the nonlinearity that distinguishes (2) from its Black–Scholes sibling (1). This means that the option trading strategies, for example, no longer could be evaluated through pricing each constituent option and then summing the prices. The financial interpretation is that in case of transaction costs the number of instruments in a portfolio matters, because it is not the same if we hedge them together or separately [8].

In case of European vanilla options, though, this problem is easily overcome. Since both call and put options have a positive gamma, (2) could be easily transformed to (1) via volatility adjustment:

$$\tilde{\sigma}^2 = \sigma^2 \left[1 - \frac{c}{\sigma\sqrt{\Delta t}} \sqrt{\frac{2}{\pi}} \right]. \quad (3)$$

Of course, $\tilde{\sigma}^2$ has to be positive, so this imposes lower bound on Δt :

$$\Delta t > \frac{2 c^2}{\pi \sigma^2}.$$

The financial interpretation of the last inequality is that it is not possible to hedge more frequently than some threshold – simply because the costs will outweigh the option value.

IMPLIED VOLATILITY

Finding the implied volatility is a fundamental problem in finances. What is more, the volatility is by far the most significant parameter influencing on the price movements of the underlying assets and hence the options. The volatility level is so important that options are quoted in terms of volatility instead of price between traders. Furthermore, the volatility is not constant but depends on the remaining time to maturity and the strike price – which is not implemented in the original Black–Scholes model [2].

There are two main approaches to find the implied volatility. The first of them is analytical. The goal of the researchers is to calculate the variation of the volatility using the market databases and the Black–Scholes formula. Usually a finite approximation of the CDF of the standard normal distribution is used as well as some simplifying assumptions – e. g. in-the-money case is considered [7].

The other main approach is based on a numerical method for finding the roots of a (nonlinear) algebraic equation. We aim at finding this value of σ , for which $f(\sigma) = C^m - g(\sigma) = 0$, where C^m is the observed market price of a call option and $g(\sigma) = V_{call}$ is the solution of (2). Such numerical algorithm is the Newton–Raphson method, which is widely spread for its simplicity and efficiency. The task for finding σ is well-posed since σ exists and it is unique in the arbitrage-free boundaries for option price $g(\sigma)$: e. g. when $g(\sigma) \in \left((S - Ke^{-r(T-t)})^+, S\right)$ for a call option and $g(\sigma) \in \left((Ke^{-r(T-t)} - S)^+, Ke^{-r(T-t)}\right)$ for a put option.

ALGORITHM

Our goal eventually is to find the (numerical) value of the implied volatility. We need the bid and ask prices of the underlying asset S_{bid} and S_{ask} as well as the bid and ask prices of the option value V_{bid} and V_{ask} , strike K , remaining time to expiration τ and the riskless interest rate r . All of them except r are directly observable from the market.

Let us denote

$$Le := \frac{c}{\sigma \sqrt{\Delta t}} \sqrt{\frac{2}{\pi}} \quad (4)$$

and then compute the implied volatility using the other parameters via the method described above. We need to emphasize that for finding σ_{bid} and σ_{ask} we use V_{bid} and V_{ask} respectively and in both procedures we input S but not S_{bid} and S_{ask} [8]. Afterwards, by solving the system

$$\begin{cases} (1 - Le)\sigma^2 = \sigma_{bid}^2 \\ (1 + Le)\sigma^2 = \sigma_{ask}^2 \end{cases} \quad (5)$$

we find the value of the unknown parameter σ . From the system (5) we derive the value of Le too and via substituting in (4) we can find the value of Δt [8].

NUMERICAL RESULTS

For the implementation R programming language is used. We access market data in real time from <http://finance.yahoo.com> employing the functionality provided in `quantmod` R package. We

use constant value for the riskless interest rate $r = 0.0183$ taken from US treasury bond rates. Next we implement the root-finding algorithm.

In order to achieve satisfactory estimate of the volatility, we need to employ a numerical method with high order of convergence and a reasonable starting point. Such a model is the Newton method [4]. For initial point we will use some of the analytical results, because they are reasonable guess and are often close to the real value. For the purpose of this paper we will make use of Bharadia, Christofides and Salkin formula [7, 2].

The described procedure is executed for every pair of τ and K . It is worth mentioning, though, that we are not guaranteed to have an option available for arbitrary expiry date and strike. To put it in other words, for some maturity the corresponding set of strikes is different from the set related to another maturity. This phenomenon makes the domain of the implied volatility surface not a rectangular but an irregular one.

We continue with exposing a part of R code responsible for retrieving the data and computing the grid of implied volatility. This particular example uses options over the stock share prices of Amazon.com Inc. for the front month:

```
Q <- 'AMZN'
Q.S <- getQuote(Q, what=yahooQF(c("Bid", "Ask")))
Sbid <- Q.S$Bid
Sask <- Q.S$Ask
S <- (Sask + Sbid) / 2
c <- (Sask - Sbid) / S
Q.opt <- getOptionChain(Q)

Vbid <- calls$Bid[i]
Vask <- calls$Ask[i]
sigma_bid <- BSImplVolCall(S, K[i], r, tau[j], Vbid)
sigma_ask <- BSImplVolCall(S, K[i], r, tau[j], Vask)
Le <- (sigma_ask^2 - sigma_bid^2) / (sigma_ask^2 + sigma_bid^2)
sigmas[j, i] <- sigma_ask / sqrt(1 + Le)
```

After gathering data and processing the implied volatilities, we have the opportunity to interpolate the values between the nodes. However, this is not a compulsory step if we use the derived values only for showing the surface and not for computational purposes. Then we plot the grid in interactive environment (Fig. 1). For the purpose of scaling we transform the functional values in their negative natural logarithm.

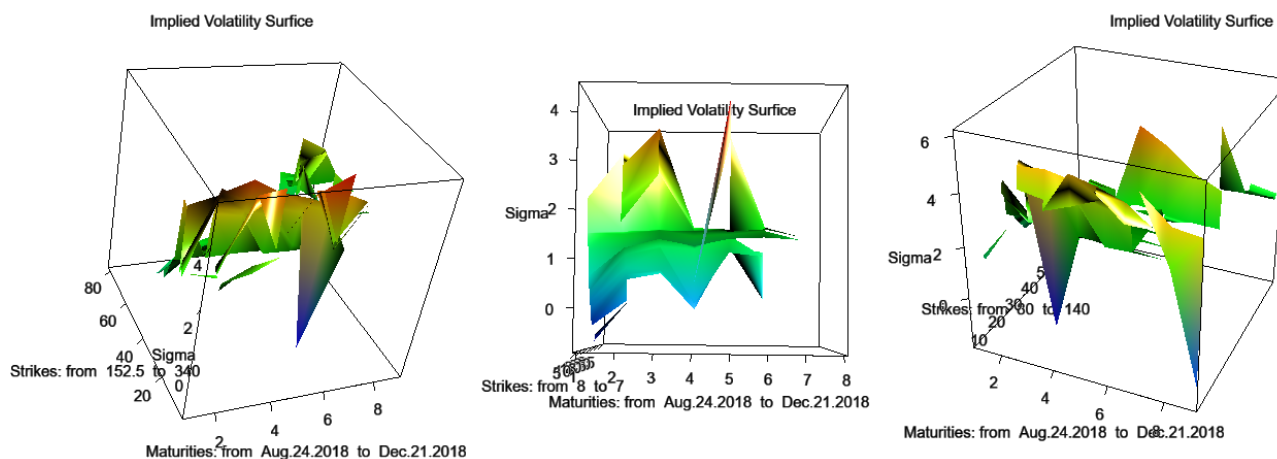


Fig. 1. Implied volatility surfaces of options on Apple Inc., General Electric Co. and Microsoft Corp. stock shares

The first feature that comes into mind is that the surface graph is to a some extent rough. This is so because we use linear interpolation between nodes, which in scarce data could be far from each other. The second reason is that the volatilities match exactly the market data which is far from being pure and clean. It is the market data that is the reason for the white holes in the surface – for such set of parameters either the option value or the value of vega is very close to zero and the numerical procedure fails to converge. In such situations the quoted volatility is reported zero as well.

A complete implementation is done using Shiny Applications and deployed to [3].

CONCLUSION

Computing the implied volatility surface is a fundamental and contemporary problem in finances. In the paper we examined calculating the surface employing the Leland model and making use of the iterative Newton method. The practical realization confirms the applicability of the algorithm.

Future developments include improvements to current model, e. g. using nonconstant interest rate or using better interpolating method, for example cubic splines. What is more, the independent variable σ could be changed to $\sigma\sqrt{\tau}$ or the target function to be logarithmised. It is possible to experiment with another method with higher order of convergence. Finally, a model other than Leland could be used.

REFERENCES

- [1] Black, F., Scholes, M. (1973). *The Pricing of Options and Corporate Liabilities*. Journal of Political Economy, 81: 637-654.
- [2] Georgiev, S. (2017). Числено пресмятане на косвената волатилност на опциите с константна еластичност на упражнителната цена по отношение на цената на базовия актив, Преддипломен проект, Русенски университет. (**Оригинално заглавие:** Георгиев, С. (2017). Числено пресмятане на косвената волатилност на опциите с константна еластичност на упражнителната цена по отношение на цената на базовия актив, Преддипломен проект, Русенски университет).
- [3] Georgiev, S. (2018). *Implied Volatility Surface Computer*. URL: <https://aslv.shinyapps.io/proj/> (Accessed in May 2018).
- [4] Jäckel, P. (2014). *Let's Be Rational*. Wilmott Journal, 75: 40-52.
- [5] Leland, H. (1985). *Option Pricing and Replication with Transactions Costs*. The Journal of Finance, 40(5): 1283-1301.

- [6] Li, S., Abdullah, M. (2012). *Implied Transaction Costs by Leland Option Pricing Model: A New Approach and Empirical Evidence*. Journal of Derivatives & Hedge Funds, 18(4): 333-360.
- [7] Orlando, G., Taglialatela, G. (2017). *A Review of Implied Volatility Calculation*. Journal of Computational and Applied Mathematics, 320: 202-220.
- [8] Stehlíková, B. (2016). *Lecture Notes on Financial Derivatives*. URL: <http://www.iam.fmph.uniba.sk/institute/stehlikova/fd16.html> (Accessed in May 2018).

ACCURATE AND NUMERICAL SOLUTIONS FOR BLACK–SCHOLES EUROPEAN OPTIONS ⁷

Nikola Borislavov – Student

Financial Mathematics

University of Ruse “Angel Kanchev”

E-mail: nikiki1994@abv.bg

Prof. Lubin Vulkov, PhD

Department of Applied Mathematics and Statistics,

University of Ruse “Angel Kanchev”

E-mail: lvulkov@uni-ruse.bg

Abstract: Accurate and Numerical Solutions for Black-Scholes European Options: The paper reviews existing differential methods. In the finite difference methods, the derivatives of the function are replaced by the corresponding difference differences. Those methods are widely used in financial mathematics because they are appropriate when the number of dimensions of the scheme is relatively small, i.e. when the function depends on a few variables. We will observe the methods applied for the heat equation and furthermore the use in Black – Scholes model.

Keywords: differential scheme, Crank–Nicolson method, finite difference method, approximations

ВЪВЕДЕНИЕ

Опцията е договор между две страни, при която едната страна е задължена, при желание на другата страна, да ѝ продаде или купи определен брой акции по предварително договорена цена и преди изтичането на определен срок. Страната, която продава опцията се нарича продавач или подписник на опцията. Страната, която купува опцията се нарича купувач или притежател на опцията. В зависимост от това, за какво се отнася опцията за покупка или продажба на акции опциите биват два вида: Кол опция - тя дава право на притежателя да закупи определен брой акции по фиксираната в опционния договор цена на притежателя си и преди изтичане на договорения срок; Пут опция – тя дава право на притежателя си да продаде определен брой акции по фиксираната в опционния договор цена и преди изтичане на договорения срок. В зависимост от това кога може да се упражни опцията се разглеждат два вида опции: Европейска – опцията може да се упражни само на фиксираната дата T ; Американска – тя може да се упражни във всеки момент от периода между издаването на опцията и датата на падежа T .

Методите на крайните разлики се използват толкова широко, защото притежават редица положителни свойства. Например те позволяват много добре да се определи реда на грешката и нивото на точност на алгоритъма, докато при други методи не е така. Още, те са удобни за пресмятане на диференциални уравнения със сравнително сложни начални и терминални условия [3].

Ние разглеждаме линеен модел на Блек–Шолс за оценка на кол и пут опция

$$\frac{\partial V}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS \frac{dV}{dS} - rV = 0$$

⁷ Докладът е представен на студентската научна сесия на 23. 05. 2018 г. в секция „Финансова математика“ с оригинално заглавие на български език: ТОЧНИ И ЧИСЛЕНИ РЕШЕНИЯ ЗА БЛЕК–ШОЛС ЕВРОПЕЙСКИ ОПЦИИ

с крайна цена, съответно за кол и пут опции

$$C(S, T) = (S - K)^+(\text{кол}) = \max(S - K, 0),$$

$$P(S, T) = (K - S)^+(\text{пут}).$$

Гранични условия най – често са следните:

$$C(0, t) = 0, C(S, t) = S - Ee^{r(T-t)} \text{ при } S \rightarrow +\infty (\text{кол}),$$

$$P(0, t) = Ee^{-r(T-t)}, P(S, t) = 0 \text{ при } S \rightarrow +\infty (\text{пут}).$$

Да положим

$$x = \ln\left(\frac{S}{K}\right), \tau := \frac{1}{2}\sigma^2(T - t), y(x, \tau) = \frac{1}{2}V\left(Ke^x, T - \frac{2}{\sigma^2\tau}\right),$$

или

$$V(x, \tau) = e^{-\frac{k-1}{2}x - \frac{(k+1)^2}{4}\tau} y(x, \tau),$$

и така намираме, че y удовлетворява уравнението на топлопроводността

$$\frac{\partial y}{\partial \tau} = \frac{\partial^2 y}{\partial x^2}, \tau \in [0, T], x \in [x_{\min}, x_{\max}].$$

От тук, връщайки се към оригиналните променливи, например за кол опцията имаме:

$$C(S, t) = SN(d_+) - Ke^{-r(T-t)}N(d_-)$$

където $N(z)$ е стандартното разпределение

$$N(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^z e^{-\frac{1}{2}y^2} dy, \quad d_{\pm} = \left(\left(\frac{x}{\sqrt{2\pi}} + \frac{1}{2} \right) k \pm 1 \right) \sqrt{2\tau}.$$

ДИФЕРЕНЧНИ АПРОКСИМАЦИИ

Диференчната апроксимация на стойността на производната на функцията $f(x)$ в точка x_0 , например $f'(x_0)$ е някаква комбинация от стойностите на функцията в точки, близки до x_0 .

Например, ако за някакво малко $h > 0$ развием $f(x + h)$ в ред на Тейлър, разделим на h и групираме членовете от двете страни на равенството получаваме

$$f'(x) = \frac{f(x + h) - f(x)}{h} + \frac{h}{2} f''(\xi)$$

за $\xi \in (x, x + h)$ и тогава

$$f'(x) \approx \frac{f(x + h) - f(x)}{h}.$$

Тъй като $f''(\xi)$ е число, то разликата между $f'(x)$ и диференчното частно е израз на h^1 , т. е. грешката, която допускаме, е от първи ред. Това означава, че ако 10 пъти намалим h , то ще получим 10 пъти по-точно приближение [2].

Диференчната разлика, която разгледахме, се нарича права или дясна разлика. Ако вземем под внимание точките $x - h$ и x вместо по-горе разгледаните, ще получим следната апроксимация:

$$f'(x) \approx \frac{f(x) - f(x - h)}{h},$$

която се нарича обратна или лява разлика. Тук грешката също е от първи ред. Ако поискаме да намалим грешката, трябва да разгледаме повече на брой точки около x . Нека вземем трите точки $x - h$, x и $x + h$. След като развием в ред на Тейлър стойностите $f(x - h)$ и $f(x + h)$, вадим първия израз от втория и записваме

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h},$$

където грешката е от втори ред, т. е. ако намалим 10 пъти h , грешката ще намалее от порядъка на 100 пъти. Това представлява централна разлика.



Фигура 1. Сравнение на права разлика (отляво) и централна разлика (отдясно) при апроксимация на първа производна.

Преимуществото на по-високия ред грешка се забелязва нагледно (фигура 1). За по-висок ред производни също трябва да изберем повече на брой точки. След подобни преобразувания получаваме централната разлика за втората производна:

$$f''(x) \approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}, \quad (1)$$

където грешката също е от втори ред.

ДИФЕРЕНЧНИ СХЕМИ

Нека разгледаме топлинното диференциално уравнение с една пространствена променлива

$$\frac{\partial y}{\partial \tau} = \frac{\partial^2 y}{\partial x^2}, \tau \in [0, T], x \in [x_{min}, x_{max}],$$

със следните гранични условия:

$$y(x, 0) = y_0(x), y(x_{min}, \tau) = y_l(\tau), y(x_{max}, \tau) = y_r(\tau).$$

Това параболично диференциално уравнение заема важно място дори във финансовата математика, тъй като уравнението на Блек и Шолс може да се сведе до топлинното само с няколко полагания.

Въпреки, че уравнение (1) има аналитично решение, ние ще разгледаме численото решаване посредством метода на Кранк – Никълсън, като за сравнение ще маркираме и другите известни диференчни схеми.

Нека дискретизираме областта $[x_{min}, x_{max}]$ на N равни интервала

$$x_i = x_{min} + ih \text{ за } i = 0, 1, \dots, N,$$

$$h = \frac{x_{max} - x_{min}}{N}$$

и нека $\tau_j = \Delta\tau \cdot j$ за $j = 0, 1, \dots, M$, $\Delta\tau = \frac{T}{M}$.

Нека приближим и $y(x_i, \tau_j) \cong Y_{ij}$.

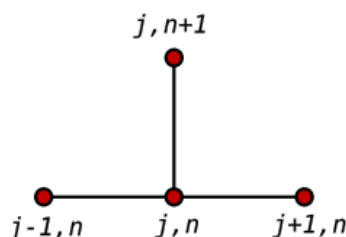
ЯВНА ДИФЕРЕНЧНА СХЕМА

При явната схема използваме правата разлика за производната по време и втората централна разлика за втората производна по пространство. След като приложим вече показаните формули получаваме

$$\frac{Y_{ij+1} - Y_{ij}}{\Delta\tau} - \frac{Y_{i-1j} - 2Y_{ij} + Y_{i+1j}}{h^2} = 0$$

и ако означим $\rho = \frac{\Delta\tau}{h^2}$, се получава

$$\begin{cases} Y_{ij+1} = \rho Y_{i-1j} + (1 - 2\rho)Y_{ij} + \rho Y_{i+1j} \\ Y_{0j} = Y_l(\tau_j) \\ Y_{Nj} = Y_r(\tau_j) \end{cases}$$



Фигура 2. Явна диференчна схема.

Структурата на възлите от мрежата, които участват в една итерация се виждат на фигура 2. Числената грешка е първи ред по време и втори по пространство:

$$\Delta y = O(\Delta\tau) + O(h^2).$$

Предимствата на явната схема са, че тя е проста и лесна за разбиране и имплементация, както и че може да се реализира итеративно, без да е нужно изчисленията за всеки времеви слой да се правят едновременно. Това води и до намалена изчислителна сложност.

Основният ѝ недостатък обаче е много значим – тя не е безусловно устойчива. За да бъде устойчива, достатъчно условие е да е изпълнено $\rho \leq \frac{1}{2}$.

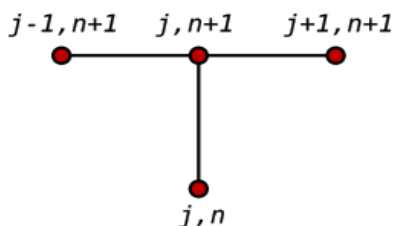
В практиката явните схеми не са често използвани.

НЕЯВНА ДИФЕРЕНЧНА СХЕМА

При неявната диференчна схема се използва обратната разлика за производната по време и отново втората централна разлика за производната по пространство. Схемата изглежда така:

$$\frac{Y_{ij+1} - Y_{ij}}{\Delta\tau} - \frac{Y_{i-1j+1} - 2Y_{ij+1} + Y_{i+1j+1}}{h^2} = 0$$

$$\begin{cases} Y_{ij} = -\rho Y_{i-1j+1} + (1 + 2\rho)Y_{ij+1} - \rho Y_{i+1j+1} \\ Y_{0j+1} = Y_l(\tau_{j+1}) \\ Y_{Nj+1} = Y_r(\tau_{j+1}) \end{cases}$$



Фигура 3. Неявна схема

Редът на грешката е същият като при явната схема: $\Delta y = O(\Delta \tau) + O(h^2)$.

Предимството ѝ пред явната схема е голямо – неявната схема е безусловно устойчива. Това, обаче, от своя страна води до повишена изчислителна сложност и съответно невъзможността изчисленията по един времеви слой да се правят последователно. Затова при всеки времеви слой се решава матрично уравнение. Матрицата, с която се работи, е с голям размер, но тъй като тя е тридиагонална, съществуват бързи методи за обръщането ѝ.

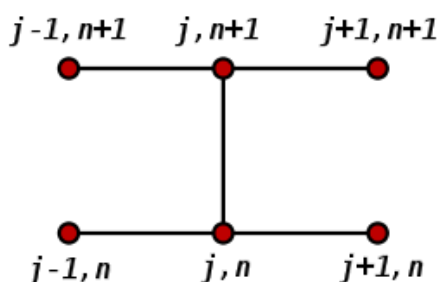
ЯВНО–НЕЯВНА СХЕМА НА КРАНК–НИКЪЛСЪН

При схемата на Кранк – Никълсън се използват централни разлики и по време, и по пространство. Това е една от основните разлики с предишните схеми. По този начин се засяга точката $Y_{ij+\frac{1}{2}}$, която не е част от мрежата. Получава се

$$\frac{Y_{ij+1} - Y_{ij}}{\Delta \tau} - \frac{1}{2} \left(\frac{Y_{i-1j+1} - 2Y_{ij+1} + Y_{i+1j+1}}{h^2} + \frac{Y_{i-1j} - 2Y_{ij} + Y_{i+1j}}{h^2} \right) = 0$$

и съответно

$$\begin{cases} 2(1 + \rho)Y_{ij+1} - 2(1 - \rho)Y_{ij} = \rho(Y_{i-1j+1} + Y_{i+1j+1} + Y_{i-1j} + Y_{i+1j}) \\ Y_{0j} = Y_l(\tau_j), Y_{0j+1} = Y_l(\tau_{j+1}) \\ Y_{Nj} = Y_r(\tau_j), Y_{Nj+1} = Y_r(\tau_{j+1}) \end{cases}$$



Фигура 4.Схема на Кранк-Никълсън

Поради централната разлика по време, при схемата на Кранк – Никълсън грешката е от втори ред и по двете измерения:

$$\Delta y = O(\Delta \tau^2) + O(h^2).$$

Схемата на Кранк – Никълсън е условно устойчива. Тогава защо се е наложило разработването ѝ? Тъй като редът на грешките и по двете променливи е еднакъв, то е разумно и двете стъпки да са от еднакъв порядък – т. е. $\Delta \tau \approx h$. Именно това дава голямото предимство на схемата на Кранк – Никълсън: методът е по-бързо сходящ и може да се постигне идентична точност спрямо другите методи с по-малък брой времеви стъпки.

Един недостатък на схемата е, че не се справя толкова добре с осцилациите, които се наблюдават около прекъсвания или ъгли при функцията на решението. Затова е добра идея първо да се използва чисто неявна схема, докато негладкостите изчезнат, и след това да се превключи към схемата на Кранк – Никълсън с по-голяма времева стъпка.

РЕАЛИЗАЦИЯ НА СХЕМАТА НА КРАНК – НИКЪЛСЪН

Диференчното уравнение може да се запише в матричен вид:

$$AY_{j+1} = BY_j + \rho(c_{j+1} + c_j),$$

където

$$A = \begin{bmatrix} 2(1+\rho) & -\rho & 0 & \dots & 0 & 0 \\ -\rho & 2(1+\rho) & -\rho & \dots & 0 & 0 \\ 0 & -\rho & 2(1+\rho) & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -\rho & 2(1+\rho) \end{bmatrix},$$

$$B = \begin{bmatrix} 2(1-\rho) & \rho & 0 & \dots & 0 & 0 \\ \rho & 2(1-\rho) & \rho & \dots & 0 & 0 \\ 0 & \rho & 2(1-\rho) & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \rho & 2(1-\rho) \end{bmatrix},$$

$$Y_j = (Y_{1j}, Y_{2j}, \dots, Y_{N-1j})^T, c_j = (Y_{0j} = Y_l(\tau_j), 0, \dots, 0, Y_{Nj} = Y_r(\tau_j))^T.$$

Очевидно решението се дава с

$$Y_{j+1} = A^{-1} (BY_j + \rho(c_{j+1} + c_j)).$$

Собствените стойности на $A^{-1}B$ са

$$\alpha_k = \frac{2 - 4 \sin^2 \left(\frac{k\pi}{2N} \right)}{2 + 4 \sin^2 \left(\frac{k\pi}{2N} \right)}, k = 1, 2, \dots, N-1.$$

Тъй като $|\alpha_k| \leq 1$ за всяко k , решение винаги съществува и схемата е условно устойчива.

Най-тежката операция в решаването на уравнението е намирането на обратната матрица на A . Въпреки, че може да се използва директно метод на Якоби, метод на Гаус – Зайдел или свръхгорна релаксация, най-добре е да се възползваме от факта, че матрицата е тридиагонална. Например може да използваме метода на прогонката.

ЧИСЛЕН ЕКСПИРИМЕНТ

Още от 1977 г. крайните диференчни схеми се използват активно в областта на финансовата математика. Досега разгледахме метода на Кранк – Никълсън, приложен върху топлинното уравнение, което от своя страна лесно се свежда до уравнението на Блек и Шолс. Обаче, последното отново е параболично диференциално уравнение и може направо към него да се подходи с крайна диференчна схема. Основният момент е да се „обърне“ времето, така че матуритетът на опцията да е в $\tau = 0$, а настоящият момент $\tau = T$. Такава програмна реализация е представена във файла на МАТЛАБ.

```
>>help optionCN
Function to calculate the price of a vanilla European
Put and Call option using the Crank-Nicolson finite difference
method
```

```
oPrice = optionCN(K,S0,r,sig,Smax,h)
```

```
Inputs: K - strike
        : S0 - stock price
        : r - risk free interest rate
        : sig - volatility
        : Smax - maximal stock price (end spatial point)
        : h - size of the spatial step
```

```
Output: oPrice - the call and put option price
```

ЗАКЛЮЧЕНИЕ

Настоящият доклад представя:

Трансформация която превежда уравнението на Блек – Шолс описващо цени на опции до топлинното уравнение. За последното са построени много числени методи. Важна стъпка от решението на една задача е правилният подбор на метод, който ще бъде използван. След направения анализ имаме възможността да сравним полезността на всеки един от методите спрямо даден проблем.

REFERENCES

- [1] Tavella, D., *Quantitative Methods in Derivatives Pricing – An Introduction to Computational Finance*, John Wiley & Sons, Inc., 2002
- [2] Olver, P., *Introduction to Partial Differential Equations*, Springer, 2014
- [3] Brandimarte, P., *Numerical Methods in Finance & Economics – A MATLAB based Introduction*, John Wiley & Sons, Inc., 2006

WED-2.203-SSS-FM-08

APPLICATION OF THE BLACK-SCHOLES MODEL FOR EVALUATING OPTIONS ⁸

Dimitar Stanev – Student

Financial Mathematics

University of Ruse “Angel Kanchev”

E-mail: mi7ko574n3v@abv.bg

Assist. Prof. Tihomir Gyulov, PhD,

Department of Mathematics

University of Ruse “Angel Kanchev”

E-mail: tgulov@uni-ruse.bg

Abstract: *Application of the Black-Scholes Model for Evaluating Options:* For years, economists and mathematicians have sought to find a convenient formula for easily calculating options premiums. In 1973 the Black-Scholes formula for the call option was published. This is the first formula to calculate the theoretical value of the premium.

Key words: *formula, Black-Scholes, options, premium.*

ВЪВЕДЕНИЕ

Във финансите опцията е ценна книга, която дава на купувача правото, но не и задължението, да купи или продаде определен базов актив или инструмент на предварително определена цена на или преди определена дата. Продавачът има съответното задължение да изпълни сделката – което е да продаде или купи – ако купувачът упражни опцията. Купувачът плаща премия на продавача за това си право. Опция, която дава право да се купи нещо на определена цена се нарича кол (call) опция. А опция, която дава право да се продаде нещо на определена цена се нарича пут (put) опция.

Всяка финансова опция е договор между две страни със срещуположни интереси. Самите договори по опциите могат да бъдат сложни, но обикновено съдържат следните характеристики: кол или пут опция, количество и вид на актива, цена на упражняване, дата на изтичане, условия по сделката, премия.

Опциите се делят на два вида: ванилови и екзотични. Ваниловите биват Американски, Бермудски и Европейски.

МОДЕЛ НА БЛЕК-ШОЛС

Моделът на Блек-Шолс е математически модел на финансовия пазар съдържащ деривативни инвестиционни инструменти. От модела може да се изведе формулата на Блек-Шолс, която се използва за пресмятане на цената на Европейската Опция. Моделът е публикуван в [1] "Journal of political economy" със заглавие "The Pricing of Options and Corporate Liabilities" от Фишър Блек и Майрън Шолс. След откриването на формулата следва драстичен ръст в търговията с опции и легитимира научно дейностите на Чикагската борса за обмяна на опции и други пазари за опции по света. Формулата на Блек-Шолс е широко използвана, но често с промени и корекции от Търговците в пазара на опции. Много емпирични тестове са показали, че цените, получени от Блек-Шолс, са приблизително близки до наблюдаваните цени, въпреки, че има някои известни несъответствия.

⁸ Докладът е представен на студентската научна сесия на 23. 05. 2018 г. в секция „Финансова математика“ с оригинално заглавие на български език: ПРИЛОЖЕНИЕ НА МОДЕЛА НА БЛЕК-ШОЛС ЗА ОЦЕНЯВАНЕ НА ОПЦИИ

Частното диференциално уравнението по модела на Блек-Шолс има вида:

$$\frac{\partial V}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} - rV = 0,$$

където: V е цената на дериватива, t е времето, което остава до падежа, σ е волатилитета на цената на опцията, S е пазарната цена на опцията, r е основния лихвен процент.

Днес формулата на Блек-Шолс е широко използвана от пазарните участници, интересувани се от търговията с опции. Тя предполага няколко ограничения:

- Опциите са от европейски тип;
- Безрисковият лихвен процент е известен и не се променя до края на срока на валидност на опцията;
- Статистически вероятностното разпределение на цената на базовия актив е логаритмично нормално;
- Транзакционните разходи се изключват;
- Акциите не изплащат дивиденди.

Формула на Блек-Шолс за пресмятане на Европейска кол опция има вида:

$$C(S_t, t) = N(d_1)S_t - N(d_2)Ke^{-r(T-t)}$$

$$d_1 = \frac{1}{\sigma\sqrt{T-t}} \left[\ln\left(\frac{S_t}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)(T-t) \right]$$

$$d_2 = d_1 - \sigma\sqrt{T-t}$$

където: C е цената на европейската кол опция, $N(d)$ е нормалното разпределение, K е страйковата цена, уговорена в договора за закупуване на опцията, $(T-t)$ е времето, което остава до падежа на опцията,

Формула на Блек-Шолс за пресмятане на Европейска пут опция има вида

$$P(S_t, t) = N(-d_2)Ke^{-r(T-t)} - N(-d_1)S_t$$

Оценката на опциите е тема както в академичните среди, така и в практиката. В основни линии, оценката на една опция е разделена обикновено на две части:

1. Вътрешната присъща стойност е неотрицателна величина, определена от разликата между пазарната стойност на базовия актив и цената на упражняване на дадена опция.

При пут опция видът ѝ е следния:

$$P = (K - S)^+ = \max\{K - S, 0\}.$$

При кол опцията:

$$C = (S - K)^+ = \max\{S - K, 0\}.$$

2. Времовата стойност зависи от съвкупност от други фактори, които чрез променлива нелинейна връзка отразяват дисконтираната очаквана стойност на датата на изтичане.

При път опциите нарастването в курса на базовия актив води до намаляване на опционната цена. Цената на упражняване е другият ключов фактор за опционната премия. Нарастването на цената на упражняване води до намаляване стойността на кол опциите. При същото движение на цената на упражняване стойността на път опциите расте. Поради обстоятелството, че притежателят на опцията не е задължен да я упражни и би го направил само в случай на икономическа целесъобразност, вътрешната стойност на една опция не може да бъде отрицателна.

Оценяването на опциите има за цел да определи тяхната теоретична, правилна стойност във всеки един момент от опционния срок. Това е сред най-важните въпроси за инвеститорите на борсовите пазари. Установяването на справедлива пазарна цена на опциите е задължителна предпоставка за успешното прилагане на търговски и портфейлни стратегии с опции. Тази безспорна значимост на проблема за оценката на опциите предизвиква и интересът към решаването му. След основополагането което прави моделът на Блек-Шолс се извеждат и други методи за оценка на опции. Въпреки различията си всички модели имат за цел пресмятане цената на опцията. Тъй като моделът на Блек-Шолс е първият с такава цел той носи и в себе си много недостатъци. По-новите модели за оценка на опции са доста по точни, а и са тясно свързани с финансовия пазар в сегашно време. Въпреки това моделът на Блек-Шолс си остава най-лесното за смятане приближение до реалните стойности на опциите.

ПРИЛОЖЕНИЕ НА МОДЕЛА НА БЛЕК-ШОЛС: РОЛОУВЪР ОПЦИЯ

Бермудската опция е деривативен инструмент, при който притежателят на опцията има правото, но не и задължението, да упражни своята опция само на предварително определени дати, настъпващи на редовни интервали през цялото времетраене на договора. Това я прави хибрид между Американска опция, при която може да се упражни по всяко време от периода на договора и Европейска опция, която може да се упражни само в края на периода на договора. Получават името си от факта, че Бермудските острови се намират географски между Съединените щати и Европа. Бермудските опции могат да бъдат път или кол.

Има няколко предимства и недостатъци в Бермудските опции. Продавачите на Бермудски опции имат по-голям контрол върху това кога опцията ще бъде упражнена. Купувачите на Бермудски опции получават вариант, който е по-евтин от Американската опция и по-малко ограничителен от Европейската опция.

Най-общо казано една Бермудска опция може да се опише по следния начин. Нека датите на упражняване са определени от времевите моменти $t_1, t_2, t_3, \dots, t_N$, $0 =: t_0 < t_1 < \dots < t_N = T$, където T е падежната дата на опцията. След това, с U_{t_i} ще означим настоящата стойност към момента $t = 0$ на платежната функция при упражняване на опцията в момента $t_i, i = 1, 2, \dots, N$. Например:

$$U_{t_i} = e^{-rt_i}(S_{t_i} - K)^+ \text{ при Бермудска кол опция с цена на упражняване } K, \text{ и}$$

$$U_{t_i} = e^{-rt_i}(K - S_{t_i})^+ \text{ при Бермудска път опция с цена на упражняване } K.$$

Нека разгледаме договор за инвестиция с гарантирана сума при падеж (maturity guarantee) (вж. Bilodeau [2]). При такъв договор е налице минимална гарантирана стойност K на инвестиция в даден базов актив към момента на падежа, тоест при падеж инвеститорът получава по-голямата от следните две стойности - гарантираната сума или пазарната цена на базовия актив. С други думи, такъв договор може да бъде разглеждан като комбинация от инвестиция в базовия актив и опция за продажба (път опция) на този актив с цена на упражняване, съвпадаща с гарантираната сума K . Нека с S_0 е означена цената на актива в началния момент $t = 0$, а с t_1 – моментът на падежа на такъв договор. Очевидно, инвестицията в такъв договор е сума от S_0 и цената на опцията за продажба. При предположенията от модела на Блек-Шоулс,

ако активът има волатилност σ , а безрисковата доходност е означена с r , то цената на договора е

$$S_0 + P(S_0, K, r, \sigma, t_1) = Ke^{-rt_1}N(-d_2) + S_0(1 - N(-d_1)) = Ke^{-rt_1}N(-d_2) + S_0N(d_1),$$

където

$$d_1 = \frac{\ln \frac{S_0}{K} + \left(r + \frac{\sigma^2}{2}\right)t_1}{\sigma\sqrt{t_1}}, \quad d_2 = d_1 - \sigma\sqrt{t_1}.$$

Ще разгледаме частен случай на Бермудска опция, т.нар. *Ролоувър опция (rollover option)* (Bilodeau [2], Schweizer [3]) върху договор за инвестиция с гарантирана сума при падеж $t_1 \in (0, T)$, където T е срокът на опцията. Тя предоставя една от следните две възможности в момента t_1 :

а) упражняване на договора, в резултат на което инвеститорът получава $\max\{S_{t_1}, K\}$;

б) подновяване на договора до следващата падежна дата с нова гарантирана сума $K \left(\frac{S_{t_1}}{S_0}\right)$.

Гарантираната сума при подновяване е определена по такъв начин, че за всеки отделен период да се запазва съотношението между гарантирана сума и цена на актива в началото му, тоест

$$\frac{\left\{ \begin{array}{c} \text{гарантирана сума} \\ \text{към момента } T \\ \hline K(S_{t_1}/S_0) \end{array} \right\}}{S_{t_1}} = \frac{\left\{ \begin{array}{c} \text{гарантирана сума} \\ \text{към момента } t_1 \\ \hline K \end{array} \right\}}{S_0}.$$

Може да се разглежда по-обща ситуация, при която има няколко междинни момента за подновяване на договора. Ние ще се спрем само на случая, когато има само един такъв момент. Можем да считаме такава ролоувър опция като частен случай на бермудска опция, чиято дисконтираната стойност на функцията на отплата е:

$$U_{t_1} = \max(X_{t_1}, Ke^{-rt_1}) = X_{t_1} + (Ke^{-rt_1} - X_{t_1})^+,$$

$$U_T = \max\left(X_T, \frac{X_{t_1}}{X_0} Ke^{-r(T-t_1)}\right) = X_T + \left(\frac{X_{t_1}}{X_0} Ke^{-r(T-t_1)} - X_T\right)^+,$$

където с $X_t = e^{-rt}S_t$, $t \in [0, T]$, е означена настоящата (дисконтираната) стойност на цената S_t на базовия актив. Ще считаме, че процесът S_t се описва чрез геометрично брауново движение с волатилност σ , а породената от него филтрация ще означаваме с $F = \{\mathcal{F}_t\}$. С P^* ще означим еквивалентната мартингална мярка, спрямо която процесът X_t е мартингал, а с $E^*[\cdot]$ ще означаваме съответното математическо очакване относно P^* . Да означим с M_{t_1} условното очакване на U_T към момента t_1

$$\begin{aligned} M_{t_1} &:= E^*[U_T | \mathcal{F}_{t_1}] = E^*\left[X_T + X_{t_1} \left(\frac{K}{X_0} e^{-r(T-t_1)} - \frac{X_T}{X_{t_1}}\right)^+ | \mathcal{F}_{t_1}\right] \\ &= X_{t_1} \left(1 + E^*\left[\left(\frac{K}{X_0} e^{-r(T-t_1)} - \frac{X_T}{X_{t_1}}\right)^+ \right]\right) \\ &=: X_{t_1}(1 + c^*), \end{aligned}$$

където

$$c^* = E^*\left[\left(\frac{K}{X_0} e^{-r(T-t_1)} - \frac{X_T}{X_{t_1}}\right)^+ \right] = \frac{K}{S_0} N(-d'_2) - N(-d'_1) \geq 0$$

$$d'_{1,2} = \frac{(\ln \frac{S_0}{K} + (r \pm \frac{1}{2}\sigma^2)(T-t_1))}{\sigma\sqrt{T-t_1}}.$$

Тук използвахме това, че $\frac{X_T}{X_{t_1}} = \exp(\sigma(W_T^* - W_{t_1}^*) - \frac{1}{2}\sigma^2(T-t_1))$ е независима от $\mathcal{F}_{t_1}$ случайна променлива, откъдето следва, че

$$E^* \left[\left(\frac{K}{X_0} e^{-r(T-t_1)} - \frac{X_T}{X_{t_1}} \right)^+ | \mathcal{F}_{t_1} \right] = E^* \left[\left(\frac{K}{X_0} e^{-r(T-t_1)} - \frac{X_T}{X_{t_1}} \right)^+ \right].$$

Тогава за настоящата стойност V_{t_1} на опцията в момента t_1 се получава

$$\begin{aligned} V_{t_1} &= \max(X_{t_1}, K e^{-rt_1}, M_{t_1}) = \max(X_{t_1}, K e^{-rt_1}, (1+c^*)X_{t_1}) \\ &= (1+c^*) \left(X_{t_1} + \left(\frac{K e^{-rt_1}}{1+c^*} - X_{t_1} \right)^+ \right), \end{aligned}$$

а за стойността или премията на опцията в началния момент $t_0 = 0$ следва, че ([2],[3])

$$V_0 = E^*[V_{t_1}] = (1+c^*) \left(\frac{K}{1+c^*} e^{-rt_1} N(-d_2^*) + S_0 N(-d_1^*) \right),$$

където

$$d_{1,2}^* = \frac{\ln \frac{(1+c^*)S_0}{K} + (r \pm \frac{1}{2}\sigma^2)t_1}{\sigma\sqrt{t_1}}.$$

ЗАКЛЮЧЕНИЕ

Дефинирахме опциите и обяснихме какво представляват те. Разгледахме техните видове, като се спряхме на Европейските кол и пут опции. След това дефинирахме модела на Блек-Шолс за пресмятането им. След това обяснихме начина по който работят Бермудските опции. Разгледахме частен случай на Бермудските опции, а именно Ролоувър опция. Изведохме формулата за оценка на Ролоувър опцията в рамките на модела на Блек-Шолс.

Теорията за цената на опциите в днешно време е призната за най-успешно развиващата се теория не само във финансите, но и измежду всички клонове на икономическата наука. През 1997 г. моделът на Блек-Шолс носи на авторите си Нобелова награда в областта на икономиката. Моделът стои като основа на по-прецизни модели. В днешно време формулата е със статута на свещено писмо в областта на финансите.

REFERENCES

- [1] Black F., Scholes M., *"The pricing of options and corporate liabilities"*, in: Journal of Political Economy, 81 (1973), 637-659
- [2] Bilodeau C. (1997), *"Better late than never: The case of the rollover option"*, Insurance: Mathematics and Economics 21, 103-111
- [3] Schweizer M. (2002) *"On Bermudan Options"* in: K. Sandmann and P. J. Schönbucher (eds.), "Advances in Finance and Stochastics. Essays in Honour of Dieter Sondermann", Springer, 257-269

COMPUTATION OF VALUE-AT-RISK, EXPECTED SHORTFALL AND MINIMUM CAPITAL REQUIREMENT FOR MARKET RISK FOR A MULTIPLE ASSET PORTFOLIO⁹

Slavi Georgiev – Student

Mathematical Modelling in Finance, Insurance and Social Work,

Faculty of Natural Sciences and Education,

University of Ruse “Angel Kanchev”, Ruse, Bulgaria

Tel.: +359 889 021 262

E-mail: aslv@abv.bg

Abstract: *In this paper an approach for improving a bank market risk framework is proposed. To begin with, the models in use are outlined and a sample portfolio is assumed. Then the corresponding risk factors are defined and the algorithm for generating scenaria is explained in detail. This approach is considered because of the nonlinearity of the portfolio. Furthermore, it has many advantages, namely the possibility of scenaria beyond those that had happened in the past and the fact that the interlinkages of the risk factors are incorporated in the empirical multivariate distribution of the changes. Next, the current and another model are used for Value-at-Risk calculation and they are compared via backtesting. Moreover, the usage of Expected Shortfall is considered and proposals for improvement are made. Finally, the computation and impact of the Minimum capital requirement are discussed.*

Keywords: *Value-at-Risk, Expected Shortfall, Minimum Capital Requirement, Market Risk, Portfolio Risk Management*

JEL Codes: *C12, C13, G10*

INTRODUCTION

Risk management is of paramount importance in contemporary world. There are number of reasons why one should invest in quantitative risk management. Every stakeholder including customers of a financial service, managers, regulators, politicians and public might have different arguments, but eventually an agreement between all the various interests should be established. For example, modern society relies on the smooth functioning of banking and insurance systems and has a collective interest in the stability of such systems. The regulatory process culminating in Basel III has been strongly motivated by the fear of systemic risk, i. e. the danger that problems in a single financial institution may spill over, and in extreme situations, disrupt the normal functioning of the entire financial system. In the face of such scenaria, society views risk management positively and entrusts regulators with the task of forging the framework for safeguarding its interests.

What is more, it is widely believed that proper financial risk management can increase the value of a corporation and hence shareholder value. The major reasons for this are that risk management can reduce tax costs and the fact that it is beneficial due to the better access to financial markets a company has than an individual investor. In addition, risk management can increase the firm value in the presence of bankruptcy costs, as it makes bankruptcy less likely [4].

In this paper we will present an approach for improving a sample bank market risk management framework. We will achieve this by comparing the currently used risk measures and models with alternative ones, evaluating their performance and suggesting replacing the current methods if needed.

STATEMENT OF THE PROBLEM

⁹ Докладът е представен на студентската научна сесия на 23. 05. 2018 г. в секция „Финансова математика“ с оригинално заглавие на български език: ПРЕСМЯТАНЕ НА СТОЙНОСТ-ПОД-РИСК, ОЧАКВАН НЕДОСТИГ И МИНИМАЛНО ИЗИСКВАНЕ ЗА КАПИТАЛ ЗА ПАЗАРЕН РИСК ЗА ПОРТФЕЙЛ ОТ МНОГО АКТИВИ.

Let us assume a bank is currently implementing the calculation of the regulatory minimum capital requirement for market risk based on Value-at-Risk methodology in accordance with Articles 362 – 369 of the Regulation (EU) № 575/2013 of the European Parliament and of the Council of 26 June 2013 on prudential requirements for credit institutions and investment firms [5]. The bank is currently calculating the minimum capital requirement for market risk using the VaR model based on CCC–GARCH(1, 1) model. Our aim is to assess the suitability of the approach via calculating the VaR with BEKK–GARCH(1, 1) model and comparing them. We will detect the better-performing model through backtesting using Kupiec and Christoffersen tests. Furthermore, we will consider replacement of VaR by the Expected Shortfall if the latter risk measure is performing better. Finally, we will calculate the impact on minimum capital requirement for market risk using both the VaR and ES approach. We will conclude the paper with suggestions and recommendations about the improvement of the market risk management framework.

For the purpose of the analysis, we will consider the following investment portfolio, although the approach works for arbitrary multiple asset (nonlinear) portfolio consisting of equities:

- € 2 000 000 invested in Procter & Gamble Co. shares (PG, in \$);
- € 2 000 000 invested in PepsiCo, Inc. shares (PEP, in \$);
- € 3 000 000 invested in Société Générale shares (GLE.PA, in €);
- € 2 000 000 invested in bitcoins (BTC).

RISK FACTORS AND DATA SOURCES

Given this portfolio and since the bank invests euros, it is exposed to the following set of risk factors:

- Price of Procter & Gamble Co. shares (in \$);
- Price of PepsiCo, Inc. shares (in \$);
- Price of Société Générale shares (in €);
- USDEUR exchange rate;
- BTCEUR exchange rate.

We will implement all the algorithms in R programming language. We will retrieve market data (on daily basis) for the risk factors from Yahoo Finance and European Central Bank using the functionality in `quantmod` and `ecb` R packages, respectively.

Due to the slight difference between working days in USA and Europe, the temporal data domains are different for the different market data. To overcome this, we do intersection of all available dates and take the data only for the common ones.

To say it again, we use the daily adjusted close price for each equity or rate. As a reliable analysis requires data for at least five years, we gather data from the beginning of 2013.

GENERATING SCENARIO

All the equities in the portfolio are risky assets, i. e. it is exposed to risk of loss. In theory we could lose everything – this is the portfolio to become worthless. This conclusion, though, is not very helpful. Therefore, we aim to find an upper limit of the loss with some probability. This is a way to define the Value-at-Risk (VaR) measure [3]. As usual, we will calculate VaR with probability level 99% and holding period one business day.

For calculating VaR we will need the changes of the risk factors. We will use the relative geometric returns defined by

$$r_t = \ln \frac{S_t}{S_{t-1}}, \quad (1)$$

which for a single day are close to their arithmetic counterparts.

There are two ways for generating scenaria for VaR calculation. One of it is the historical simulation. After calculating the returns for some period of time, then we define a window size. In every window we have observations which define the empirical cumulative distribution function. Then VaR is equal to the resp. quantile of that empirical distribution.

The main advantages of this method are the simplicity of the implementation and the fact that it does not require any specific assumption on the class of the distribution function. The main issue, though, is the assumption of stationarity [4]. As we examine the plot of the returns (fig. 1), we can discover that this is not the case, so it appears that historical simulation does not fit our needs.

Another approach is to use a parametric method. In our case it is best to use such with an assumption of non-stationary distributions. Another advantage is that one is not limited only to cases that had happened in the past. The bank currently uses CCC–GARCH(1, 1) model, which could be defined as follows:

$$\begin{aligned} r_t &= c + \varepsilon_t, \quad \varepsilon_t \sim N(0, \Sigma_t), \\ r_t &= (r_t^1, \dots, r_t^5)^T, c = (c^1, \dots, c^5)^T, \varepsilon_t = (\varepsilon_t^1, \dots, \varepsilon_t^5)^T, \\ \Sigma_t &= \text{diag}(\sigma_t) R \text{diag}(\sigma_t), \quad \sigma_t = (\sigma_t^1, \dots, \sigma_t^5)^T, \\ (\sigma_t^i)^2 &= \omega_i + \beta_i (\sigma_{t-1}^i)^2 + \alpha_i (\varepsilon_{t-1}^i)^2, \quad i = 1, \dots, 5, \end{aligned} \quad (2)$$

where R is the stationary correlation matrix, which can be computed as the (unconditional) variance–covariance matrix of standardised residuals. In CCC–GARCH model it is assumed that volatilities are time-dependent but correlations between factors remain constant [3].

We implement model (2) with the help of `ccgarch` package. For the coefficients to be estimated, we have to provide initial values. They are derived from five univariate fits (for every risk factor) with `fGarch` package. Then a number of scenaria is generated for each particular day.

The challenger model is called BEKK–GARCH(1, 1) and is defined as follows:

$$\begin{aligned} r_t &= c + \varepsilon_t, \quad \varepsilon_t \sim N(0, \Sigma_t), \\ r_t &= (r_t^1, \dots, r_t^5)^T, c = (c^1, \dots, c^5)^T, \varepsilon_t = (\varepsilon_t^1, \dots, \varepsilon_t^5)^T, \\ \Sigma_t &= C^T C + A^T \Sigma_{t-1} A + B^T \varepsilon_{t-1} \varepsilon_{t-1}^T B, \end{aligned} \quad (3)$$

where A , B and C are square matrices of parametres. It allows for interaction between asset returns and volatiltites. Both (2) and (3) preserve the positive semi-definiteness of the covariance matrix [1].

We can implement (3) making use of `mgarchBEKK` package. However, if we do this in a straightforward manner, we encounter a limitation. The estimator has to fit 65 parametres for 5-dimensional data, which is not feasible for reasonable time. One approach to tackle the situation is to use dimation reduction technique, e. g. principle component analysis. We choose the threshold of two principal components by the criteria in [3] – for example, when estimating the whole data, the first and second components bring 86% and 10% of the variability. What is more, only the first eigenvalue is greater than the average. After estimation, we do backward transformation to the original problem.

As we noted before, there are periods with low volatility and periods with high volatility (fig. 1). In this case a model, which is calibrated once on all data, would perform poorly. To overcome this, we gradually recalibrate the models. We do recalibration on every month. The first recalibration, however, is done after more than a year – this is because the numerical procedures for fitting do not converge for shorter period.

The number of scenaria per day is chosen more or less empirically. We generate 1000 scenaria per day and this appears to be enough, because the difference in the results compared to the approach with 10000 scenaria per day is insignificant.

So far we generated “hypothetical” scenaria for tomorrow values of the risk factors. Using them we generate “hypothetical” tomorrow prices for all equities. For each scenatio, we calculate “hypothetical” tomorrow value of the whole portfolio as a weighted average of the prices of the equities. For each scenario, we calculate the relative changes in the portfolio value in a way similar

to (1). These constitute the empirical distribution of the relative changes, where the respective quantile is calculated from, which in turn is used for computing the relevant risk measure.

CALCULATING VALUE-AT-RISK AND BACKTESTING

After generation of the scenaria, we continue with the calculation of VaR itself. This is done just as taking the respective quantile of the relative empirical distribution. We work with confidence 99%, so we take the 1-percentile (with opposite sign) [3]. The performance of both the models could be observed on fig. 1.

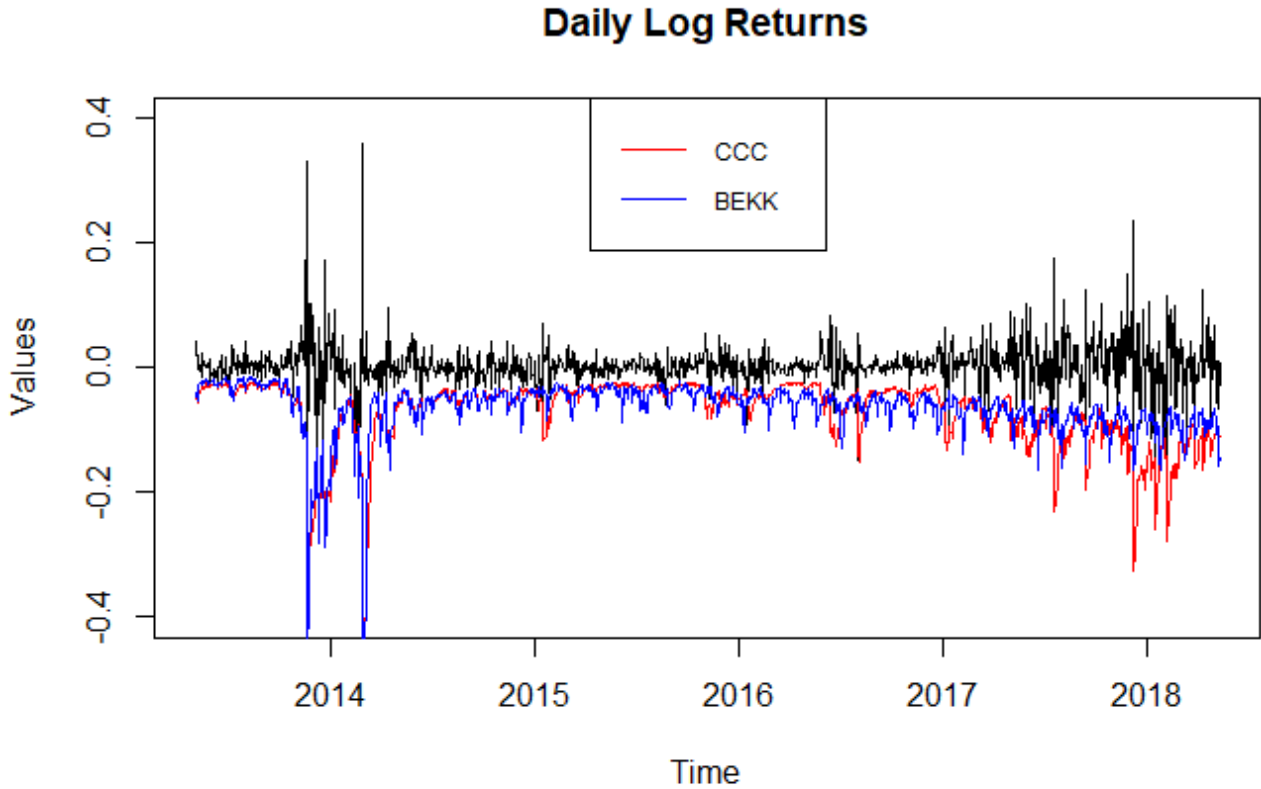


Fig. 1. Daily logarithmic returns with VaR, estimated by CCC–GARCH(1, 1) and BEKK–GARCH(1, 1) models resp.

Now we apply backtesting procedures. Two powerful tests are these of Kupiec and Christoffersen. The first of them is a statistical test that examines whether the proportion of failures calculated from the data corresponds to its theoretical value. Hence, it is focused on the unconditional coverage property only [3].

We observe VaR for N days experiencing K exceedances. We adopt the null hypothesis $\mathcal{H}_0$ that the actual relative frequency of exceedances q corresponds to the chosen probability level p . The test is a likelihood ratio test and it is based on the following statistics:

$$LR_{POF} = -2(\ln \Lambda(K, N, p) - \ln \Lambda(K, N, p)), \quad (4)$$

where

$$\Lambda(K, N, \theta) = \binom{N}{K} \theta^K (1 - \theta)^{N-K}.$$

The LR_{POF} statistics is asymptotically χ^2 distributed with one degree of freedom. We accept $\mathcal{H}_0$ with 95% significance if LR lies in the non-rejection interval for the respective quantile and number of observations.

Unlike (4), the second test we will apply focuses on both frequency and independence of the exceedances. The statistics follows:

$$LR_{Ch} = LR_{PoF} + LR_{ind}, \quad (5)$$

where

$$LR_{ind} = -2 \ln \left(\frac{(1-q)^{N_{00}+N_{10}} q^{N_{01}+N_{11}}}{(1-q_{01})^{N_{00}} (1-q_{11})^{N_{10}} q_{01}^{N_{01}} q_{11}^{N_{11}}} \right),$$

where

N_{01} – the number of days with an exceedance following after no exceedance,

N_{11} – the number of days with an exceedance following after an exceedance,

N_{00} – the number of days with no exceedance following after no exceedance,

N_{10} – the number of days with no exceedance following after an exceedance,

q_{01} – relative frequency of exceedances given no exceedance occurred the day before,

q_{11} – relative frequency of exceedances given an exceedance occurred the day before,

$$q_{01} = \frac{N_{01}}{N_{00} + N_{01}}, \quad q_{11} = \frac{N_{11}}{N_{10} + N_{11}}, \quad q = \frac{N_{01} + N_{11}}{N_{00} + N_{01} + N_{10} + N_{11}}.$$

The LR_{Ch} statistics is asymptotically χ^2 distributed with two degrees of freedom. Apparently, if $q \approx q_{01} \approx q_{11}$, LR_{ind} is close to zero. When $p \approx q$ in (4), LR_{PoF} is close to zero and we do not reject $\mathcal{H}_0$.

The values (4), (5) for our portfolio are as follows: $LR_{PoF} = 11.17$ and $LR_{Ch} = 11.49$ for CCC–GARCH(1, 1) model and $LR_{PoF} = 43.37$ and $LR_{Ch} = 46.88$ for BEKK–GARCH(1, 1) model. We easily conclude that LR_{PoF} and LR_{Ch} lie in the non-rejection interval [6, 20] for CCC–GARCH(1, 1) but not for BEKK–GARCH(1, 1) [2]. Not only we accept the first and reject the second model, but we suggest the bank continue to use CCC–GARCH(1, 1) as so far.

CALCULATING EXPECTED SHORTFALL

Although VaR is probably the most spread risk measure for market risk, it suffers from two major drawbacks. First of all, VaR represents the minimum loss given we are in market turmoil, but it does not contain information about the magnitude of losses exceeding it. Secondly, VaR is not a subadditive risk measure. For a proper risk measure it is reasonable to expect that the sum of risks of several parts of a portfolio does not exceed the risk of the whole portfolio (the effect of the diversification). However, this might not be true for VaR [3].

This is why another risk measure is introduced – Expected Shortfall. ES is defined as the expected loss given that it is greater than VaR. Sometimes it is called conditional VaR. It is a subadditive risk measure.

Taking the recommendations of the Basel Committee into account, we will compute ES at probability level 97.5%. Then we will compare it with the aforementioned results via backtesting. For ES calculating we use the established CCC–GARCH(1, 1) model. Applying ES we derive $LR_{PoF} = 0.98$ and $LR_{Ch} = 1.31$. Obviously, these values are very close to zero and we accept the null hypothesis. At first glance the usage of ES may look too conservative, but this is not the case (fig. 2). The differences between the values of VaR and ES are small and the number of exceedances is the same – 26. Nevertheless, based on the vast difference in the test statistics, we strongly recommend the bank switch to ES when using risk measure for market risk.

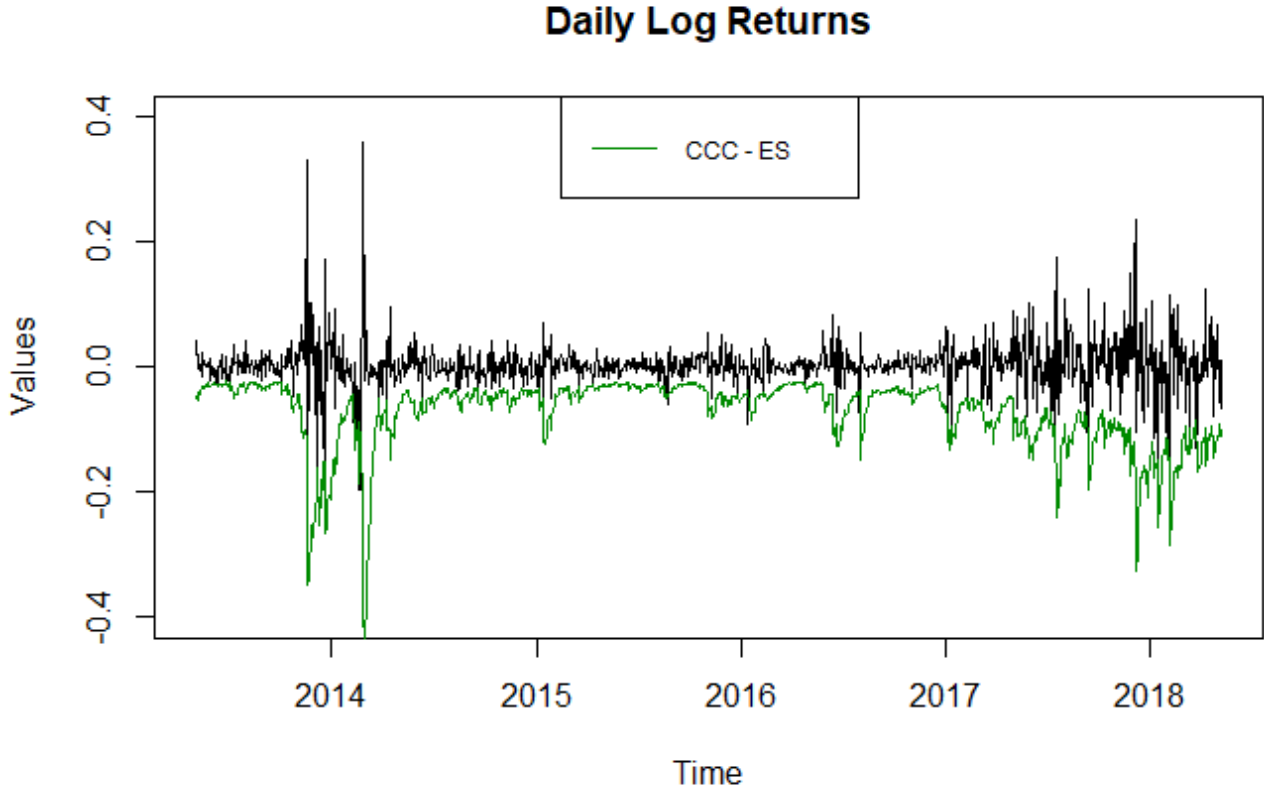


Fig. 2. Daily logarithmic returns with ES, estimated by CCC–GARCH(1, 1) model

STRESSED RISK MEASURES AND MINIMUM CAPITAL REQUIREMENT

VaR has many important applications in financial management. For example, VaR is used in internal limit systems in asset management, e. g. for imposing limit sets for individual traders. The main usage, though, is for determining the minimum capital requirement. VaR can be interpreted as the minimum amount of capital to cover losses with a predefined probability level. The requirement to hold this capital is set by the bank internal management, supervising authorities or investors and credit rating agencies [4].

The minimum required capital, in accordance with [5], is calculated in the following way:

$$MRC_t = \max \left\{ VaR_t^{10}(99\%), \frac{m_c}{60} \sum_{i=1}^{60} VaR_{t-i}^{10}(99\%) \right\} + \max \left\{ sVaR_t^{10}(99\%), \frac{m_s}{60} \sum_{i=1}^{60} sVaR_{t-i}^{10}(99\%) \right\}, \quad (6)$$

where m_c and m_s are bank-specific multipliers prescribed by the regulator based on the model quality and whose values depend on the simple regulatory test. Furthermore, sVaR is called stressed VaR and is calculated for the current portfolio as calibrating the model to data from a continuous 12-month period of significant financial stress relevant to the bank's portfolio.

For the hectic period we choose the last two years based on fig. 1 and 2. The calculation of sVaR is done in the same way as VaR except the fact that the model is calibrated with the data from the middle of 2016.

What is more, we need to use VaR with 10-day holding period. The common practice is the usage of multiplicative scaling factor of $\sqrt{10}$, i. e. $VaR^{10} = \sqrt{10} VaR^1$. This, however, is only correct when the returns follow a geometric random walk (based on normal distribution) with zero mean.

As we did previously, we repeat these calculations using ES at probability level 97.5% instead of VaR at 99%. We find that within the most recent 250 business days each of VaR and ES encounter 8 exceedances, while each of sVaR and sES encounter 7 exceedances. Thus $m_c = 3.75$ and $m_s = 3.65$. The values of MRC (6) are plotted on fig. 3:

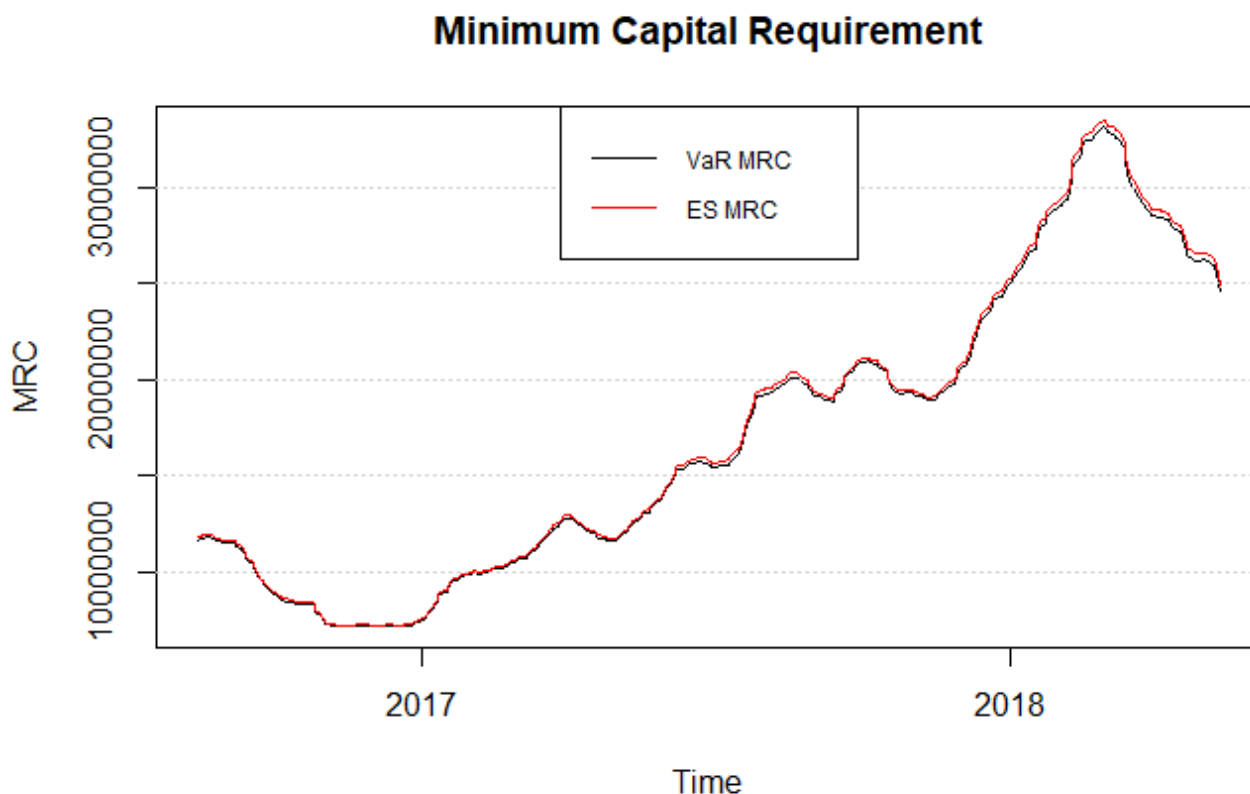


Fig. 3. Minimal required capital, estimated with VaR 99% and ES 97.5%

It is obvious that in hectic periods MRC is greater than in periods with low volatility. Furthermore, in relative units, the difference between MRC calculated by VaR and ES is not huge. Despite this, because of the importance of these values, we recommend using ES for calculating the minimum capital requirement.

CONCLUSION

As we stated in the beginning, risk management is a vital part of the modern society. In its world there are no right models but we have to assess the pros and cons of the various models by exploring its assumptions and traits. However, we must be aware that all models simplify the reality and they impose deviation we cannot escape from. For example, as we compute VaR we cannot be 99% confident that the loss wouldn't exceed VaR when we liquidate the position, e. g. due to the market illiquidity. Nevertheless, a thorough analysis of the used models can reduce the risk and bring confidence in the market risk framework.

A sample way to extend this research is to consider other risk measures, e. g. lower and upper partial moments, or assume the returns belong to a different class (heavy-tailed) distributions, for example Student t distribution.

REFERENCES

- [1] Danielsson, J. (2011). *Financial Risk Forecasting: The Theory and Practice of Forecasting Market Risk, with Implementation in R and Matlab*. Chippingham: John Wiley & Sons Ltd.
- [2] Holton, G. *Value-at-Risk: Theory and Practice*. Second Edition. URL: <https://www.value-at-risk.net/backtesting-coverage-tests/> (Accessed in May 2018)
- [3] Jurča, P. (2018). *Lecture Notes on Quantitative Methods in Risk Management*. Bratislava: Comenius University.
- [4] McNeil, A., Frey, R., Embrechts, P. (2005). *Quantitative Risk Management: Concepts, Techniques and Tools*. Princeton: Princeton University Press.
- [5] Regulation (EU) № 575/2013 of the European Parliament and of the Council of 26 June 2013 on prudential requirements for credit institutions and investment firms. URL: <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32013R0575> (Accessed in May 2018)

TERMOHUB – IOT WEB SERVICE¹⁰

Borislav Kosharov– Student

Department of Informatics and Information Technologies,

University of Ruse “Angel Kanchev”

E-mail: bosakmaw@gmail.com

Abstract: Termohub is a web application for managing internet of things devices, storing data from their sensors and displaying charts with the collected data.

Keywords: Internet of Things, web application

INTRODUCTION

The release of ESP8266 [1] device a few years ago enabled internet connectivity of devices and a new area of IT has been created – Internet of Things (IoT). The price of that development board is approximately \$2 and it has WiFi and a full network stack. Termohub is a web application developed in C# and ASP .NET Core [2]. It was first created only for temperature sensors (hence the name), but now any kind of sensor can be used. For the database I used Microsoft SQL Server [3] since it integrates well with ASP .NET Core and Entity Framework Core. Entity Framework is a ORM (object-relational mapper) framework for .NET Core and was used to design the database using code-first approach. Since the whole application is done with .NET Core it means it is cross-platform and can be run in Windows, Mac OS X and GNU/Linux. The firmware for the IoT devices was written in C++ in the Arduino [4] environment.

A lot of other IoT-related applications exist, but most of them are paid and offered as a cloud service. Termohub is free and open-source [5] and offers a DIY (Do it yourself) approach. It can either be started on your own machine or just use it on my server [6].

ARCHITECTURE

1. System design

The whole system can be separated in these modules: Firmware, Web API, Database, Web app. Figure 1 shows how all of the modules interact together.

The IoT device sends the data it collects from sensors every N seconds (can be changed from the web interface) in JSON format in a POST request to the server. The server handles those requests and stores them in SQL Server. Later an user can access that data trough the web site in the form of charts and can also configure each device and sensor. Next I will explain each module in more detail.

¹⁰ Докладът е представен на студентската научна сесия на 03.05.2018 г. в секция „Информатика“ с оригинално заглавие на български език: TERMOHUB – IOT УЕБ УСЛУГА.

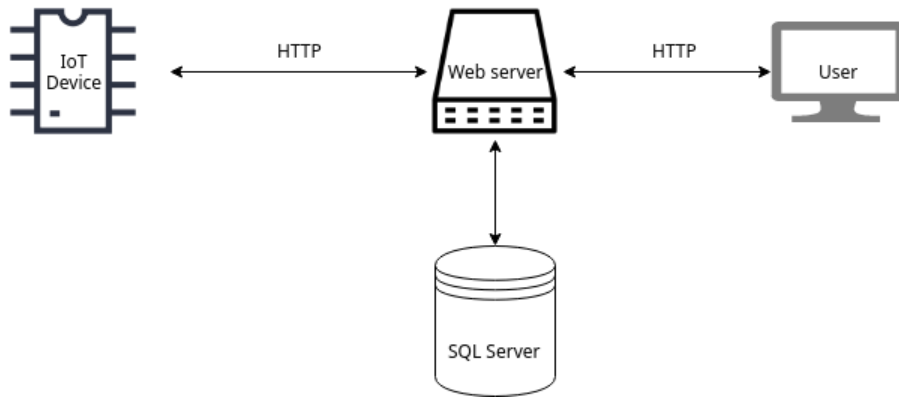


Figure 1: Architecture

2. Database

The database was designed in code-first approach which means no SQL code was written. The tables were designed as C# classes where each field is a column in the database. All the classes are mapped to tables with Entity Framework. It also translates LINQ (Language INtegrated Query) to SQL queries. This is very beneficial since SQL has different dialects and if you want to change the database engine you will have to change the SQL queries too, but with entity framework you write the table models and queries in C# and it translates them to SQL of the database you use. Any other Entity Framework supported database can be used. The only change you have to make is change the connection string.

Entity Framework also supports migrations. This means that the database model can have versions and we can change the design of the database iteratively without losing any data. Entity Framework generates these migrations for us based on our model C# classes. So to add a column to a table is as easy as defining a new field in a class and tell it to generate a new migration.

The actual database model is designed like this. There is the User table that contains a list of Devices that the user owns. Each Device has columns for Name, Delay (this is the delay between device requests), Owner (an user) and a list of Sensors that belong to this Device. Each Sensor has a Name, Unit (unit of measure °C/°F used for display only), an optional Alert and a list of Readings. A Reading has Time and Value and represents a single data point in time. The Sensor's Alert stores Sign (representing > or <), Limit (a threshold number), Email (to who to send the alert) and IsNotified (if an e-mail was already sent). The database diagram is shown on Figure 2.

3. Firmware

The device firmware is written in C++ using the Arduino environment. It uses several open-source libraries for Arduino and ESP8266. It runs in two modes: configuration and execution.

When the device first starts it checks if it has a WiFi configuration stored in its EEPROM. If it doesn't have then it turns itself into AP (Access Point) with SSID "Termohub" and password "12345678". It also starts a DNS server and a web server. When the user connects to the AP the device redirects all requests to its web server. An interface opens and lets the user configure WiFi by choosing SSID and entering a password for his own router. The user also has to enter his credentials for his account in Termohub's web service (you have to register beforehand). When the device connects successfully it stores the configuration in its EEPROM. The next time it boots it would not turn itself into AP, but it will connect to the configured router as a client.

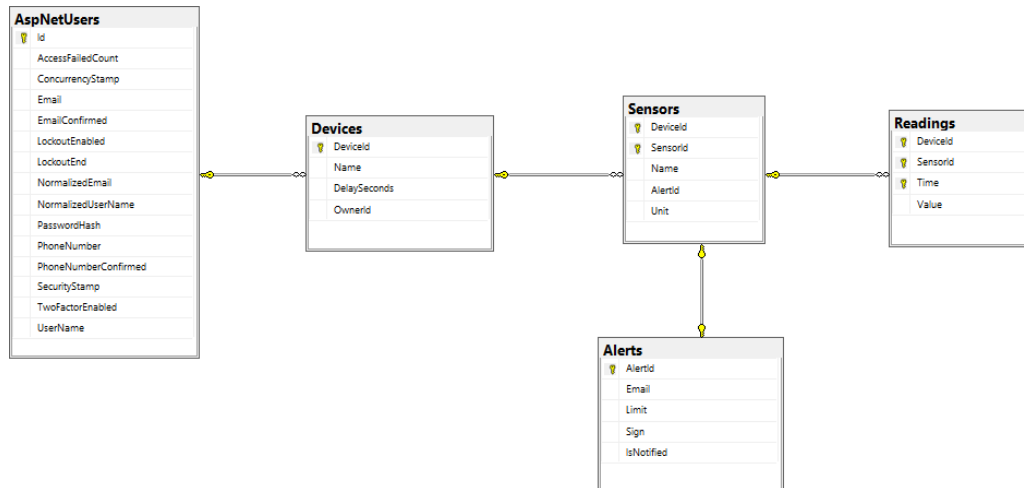


Figure 2: Database relations

In the execution mode the device first sends a request to the web API with the stored user credentials and if successful gets an access token. Then the device initializes all sensors using the OneWire protocol. This protocol can be used to connect many sensors on a single bus and only use 3 pins for the whole bus. Each sensor has an address built-in. The protocol also enables connecting a different number of sensors without changing the code. It automatically finds all sensors on the bus. After initialization the device enters the main loop. It reads each sensor's value and builds a JSON payload. It sends a DeviceID (formed by taking a part of the device's MAC address), SensorID (formed by taking two bytes of the OneWire address) and Value (the reading from that sensor). Note that no Time value is sent, because it is the time the server writes the value in the database. As a response the server returns a single number that represents the delay the device should sleep before it's next request. With each request an Authorization header with the access token is also sent. This means that all unauthenticated requests will be ignored by the server.

WEB APPLICATION

ASP .NET Core is organized in MVC (Model View Controller) architecture and has DI (Dependency Injection) built-in. DI allows any service a controller needs to be injected into the constructor of that class. The models are the classes mentioned in the Database section used to design the tables. The controllers handle HTTP requests and serve the views to the user. The views contain the HTML content, but may contain some C# logic.

Users can be in Admin or User role. The Admins can view all the devices and sensors in the system while normal Users can view only those who they own. There are two types of authentication schemes: Cookie and JWT. Cookie authentication is used when using a browser while the JWT is for the API for the devices. The Account controller handles user registration and login. The first user that is registered in the system becomes an Admin. Each next user becomes a normal User, but Admins can promote normal users to Admins.

The Device controller shows a list of devices (Figure 3). If the current user is an Admin it shows all devices of all users. If the user is a normal User it only shows the devices that user owns.



Figure 3: List of devices

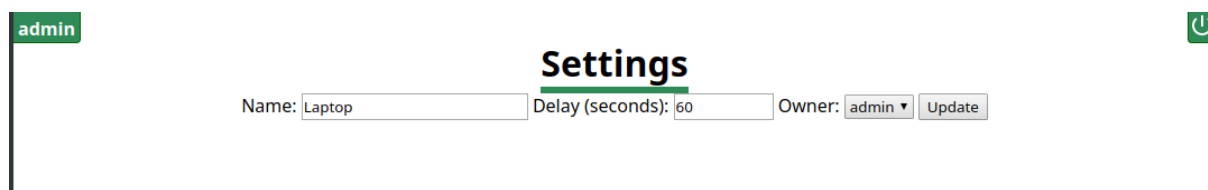


Figure 4: Device settings

Each device can be configured (by pressing on the cog). When configuring a device it's name and delay can be changed. If the current user is an admin then also the device's owner can be changed (Figure 4).

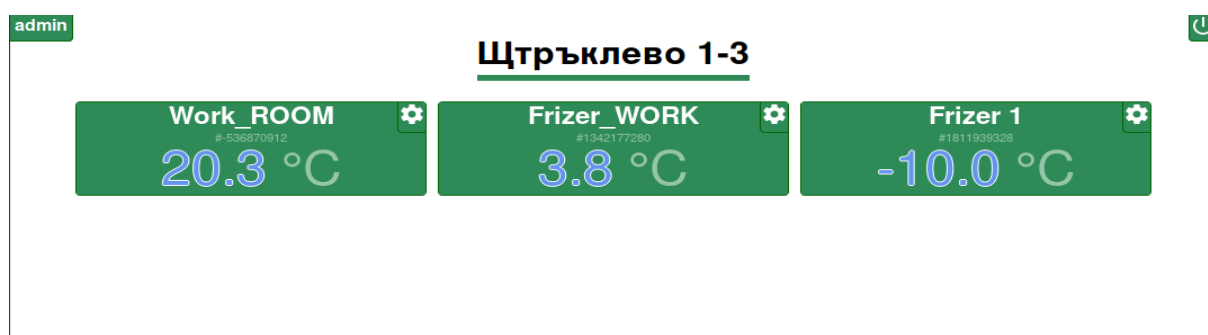


Figure 5: List of sensors

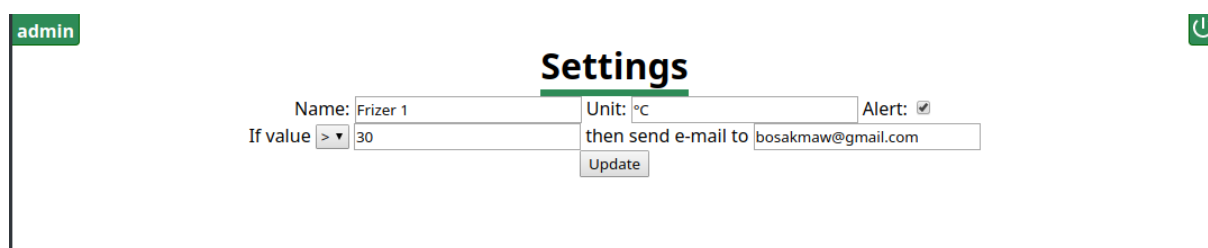


Figure 6: Sensor settings and alert

When you click on a device a list of that device's sensors is shown. The user interface is the same as the list of devices. For each sensor the most recently recorded temperature is shown (Figure 5).

Each sensor can be configured to set it's name, unit of measure and set an alert (Figure 6). When the alert checkbox is checked the additional fields bellow appear. When there is a reading with value above/below the given limit value an e-mail will be sent with an alert and a link to that sensor. An e-mail is sent only once when the limit value is exceeded. Another e-mail is sent when the temperature goes back to normal confirming that the problem is fixed.

When you click on a sensor a page with a chart with the most recent temperatures is drawn. The same page can be used to show information for an older date given you provide from and to dates interval. The Live link changes the view to a similar chart that updates itself live as data comes in. The report link downloads a .csv format file with the data in the from-to date interval. It can later be used in a spreadsheet program.

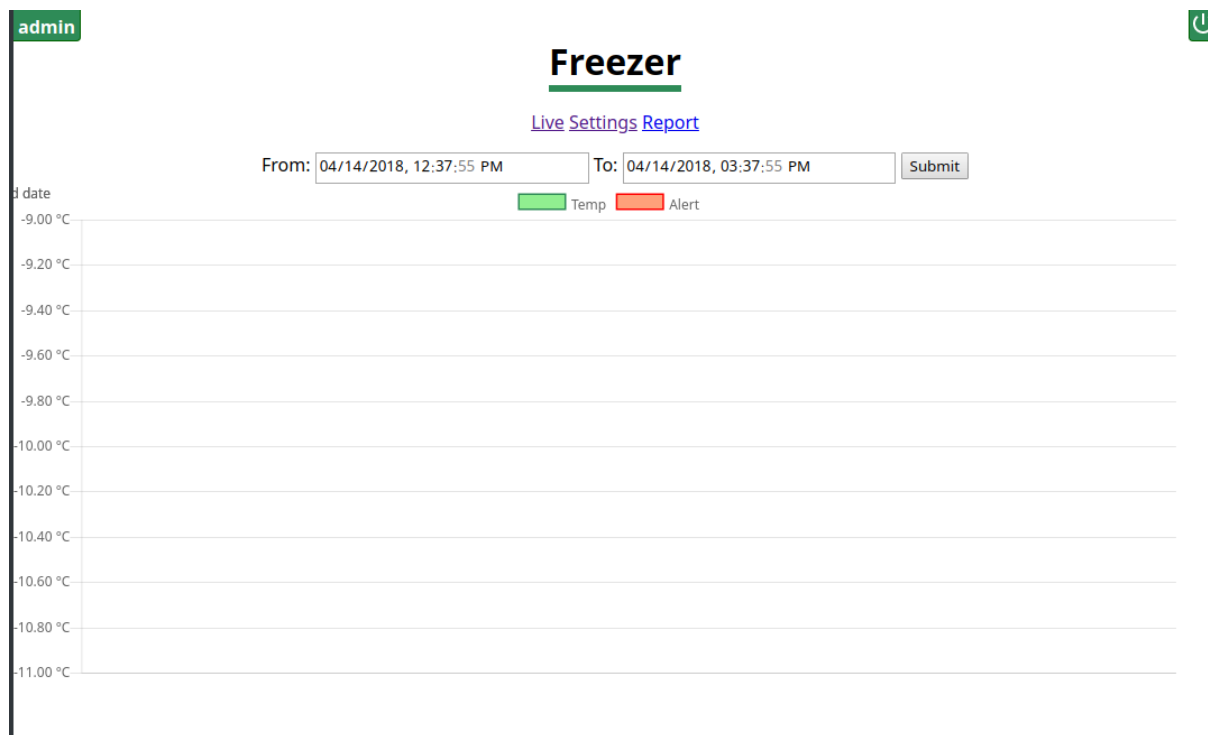


Figure 7: Sensor details and chart

The front-end part of the application that draws the charts is done in JavaScript using the Chart.js library. For the live chart every few seconds the script requests for newer data and appends it to the chart points.

CONCLUSION

Developing for the internet of things devices allows us to use existing technologies for web development to do exciting projects. Devices are now connected to the internet and we can think about them as web services. Termohub involved full stack programming from doing the hardware, writing the firmware, doing the back-end web API and then doing the front-end charts. I hope this paper has given some insight in this new field of IT.

Acknowledgement: This work was supervised by Prof. *Katalina Grigorova* from *University of Ruse*.

REFERENCES

- [1] ESP8266 device <https://en.wikipedia.org/wiki/ESP8266>
- [2] ASP .NET Core framework - <https://www.asp.net/>
- [3] Microsoft SQL Server - <https://www.microsoft.com/en-us/sql-server/sql-server-2017>
- [4] Arduino - <https://www.arduino.cc/>
- [5] Termohub on Github - <https://github.com/nikibobi/termo-hub>
- [6] Termohub deployed - <http://asp.ru-se.com:5000>
- [7] Chart.js library - <https://www.chartjs.org/>

WEB-BASED TRANSPORT MANAGEMENT SYSTEM¹¹

Tihomir Vassilev – Student

Department of Informatics and Information Technologies,
University of Ruse “Angel Kanchev”
E-mail: s126605@stud.uni-ruse.bg

Assist. Prof. Magdalena Andreeva, PhD

Department of Informatics and Information Technologies,
University of Ruse “Angel Kanchev”
Phone: +359 82 888 470
E-mail: mhandreeva@uni-ruse.bg

Abstract: The paper discusses the development of a web-based system for transport management. A comparison of existing ones is done. The choice of technology and programming language is motivated.

Keywords: Web-based system, Transport management system

ВЪВЕДЕНИЕ

Web-базираните системи намират голямо приложение в изпълнението на множество рутинни дейности. Интернет предлага голямо количество информация, която е леснодостъпна по всяко време, от всяко място и с всяко устройство, свързано към глобалната мрежа. Статията описва проекта и реализацията на **Web-базирана система за управление на автопарк**. За целта е създадено платформено независимо уеб приложение, състоящо се от модул за клиент, модул за мениджър, модул за шофьор и административен модул.

СИСТЕМИ ЗА УПРАВЛЕНИЕ НА ТРАНСПОРТ - СРАВНЕНИЕ

Съществуват много системи за управление на транспортна дейност. Основните процеси в тази дейност са – Планиране, Извършване на транспорта, Проследяване и Измерване с цел калкулация на стойност. Преглед на най-използваните, към март 2018 г. системи може да бъде намерен в <https://www.capterra.com/transportation-management-software/> (*Transportation Management Software*), а сравнение на три от тях е представено в Таблица 1 (*Comparison of Transport Management Systems*)

Таблица 1

	 DispatchMAX by First BIT Canada ★★★★★ 5 / 5 (5)	 MyRouteOnline by MyRouteOnline ★★★★★ 5 / 5 (20)	 Transporters.io by Transporters Interna... ★★★★★ 5 / 5 (1)
Best for			
Who uses this software?	trucking companies, 3PL logistic companies.	sales calls, deliveries, maintenance, trucking, courier, recycling, inspections, insurance, real estate, laundry, snow plowing, landscaping, pool cleaning, newspapers, pest control	Transport operators who offer pre-booked private hire. Coach/Bus hire and charter, pre-booked taxis, airport transfers and much more. Suitable for both vehicle operators as well as brokers.
Target Customer Size (Users)	2 - 999	1000+	50-99
Pricing			

¹¹ Докладът е представен на студентската научна сесия на 03.05.2018 в секция „Информатика“ с оригинално заглавие на български език: УЕБ БАЗИРАНА СИСТЕМА ЗА УПРАВЛЕНИЕ НА АВТОПАРК.

Starting Price	\$130.00/month/user	\$14.95/one-time	\$19.00/month/user
Free Trial	✓ YES	✓ YES	✓ YES
Free Version	✓ YES	✓ YES	✓ YES
Product details			
Platform	Web/Windows iOS/Adroid	Web Mobile - NO	Web Android
Features	Billing & Invoicing Carrier Management Customer Management Dispatch Management Driver Management Fleet / Equipment Management Load Management Order Management Quotes / Estimates Routing Scheduling Shipment Management	Not provided by vendor	Billing & Invoicing - Customer Management Dispatch Management Driver Management Fleet / Equipment Management - Order Management Quotes / Estimates Routing Scheduling -
Ratings			
Overall	★★★★★ 5/5 (5)	★★★★★ 5/5 (20)	★★★★★ 5/5 (1)
Ease of Use	★★★★★ 5/5	★★★★★ 5/5	★★★★★ 5/5
Customer Service	★★★★★ 5/5	★★★★★ 5/5	★★★★★ 5/5
Features & Functionality	N/A	★★★★★ 5/5	★★★★★ 5/5
Value of Money	N/A	★★★★★ 5/5	★★★★★ 5/5
Training & Support			
Support	- Business Hours Online	24/7 (Live Rep) Business Hours Online	- - Online
Training	In Person Live Online Webinars Documentation	In Person Live Online - Documentation	In Person Live Online - Documentation

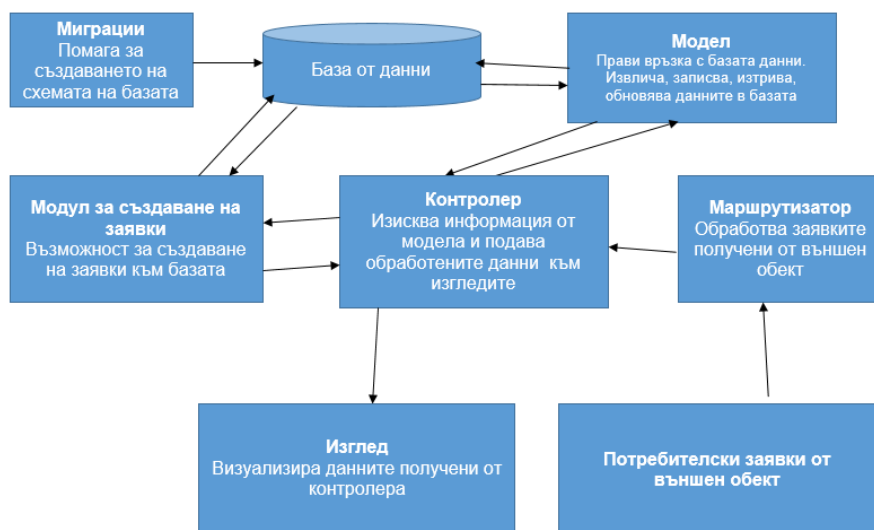
В резултат на направеното сравнение, може да се обобщи, че тези системи с предназначение за големи автопаркове и за големи товари и не са подходящи за малък бизнес.

АРХИТЕКТУРЕН МОДЕЛ НА СИСТЕМАТА

1. Избор на технология за разработка и програмен език

За реализацията на системата са използвани едни от най-разпространените и използвани в момента технологии - PHP, MVC шаблон за разработване на софтуер-Laravel 5.4, MySQL, HTML5, CSS3, JavaScript, JQuery 1.8.3, GIT и Composer. Тяхно съществено предимство е това, че са безплатни и общодостъпни, поддържат се от различни платформи и браузъри.

За функцията „Избор на маршрут“ се използва Google Maps API. Това е популярна картографска система, която предоставя географски карти на целия свят. Google Maps предлага различни маршрути с подробни упътвания за придвижване с автомобил, велосипед или пеша. Информацията постоянно се актуализира и допълва от клиентски отзиви за хотели, ресторанти и други обекти. Google Maps може да бъде вграден в уебсайт и се използва безплатно (Google Maps API).

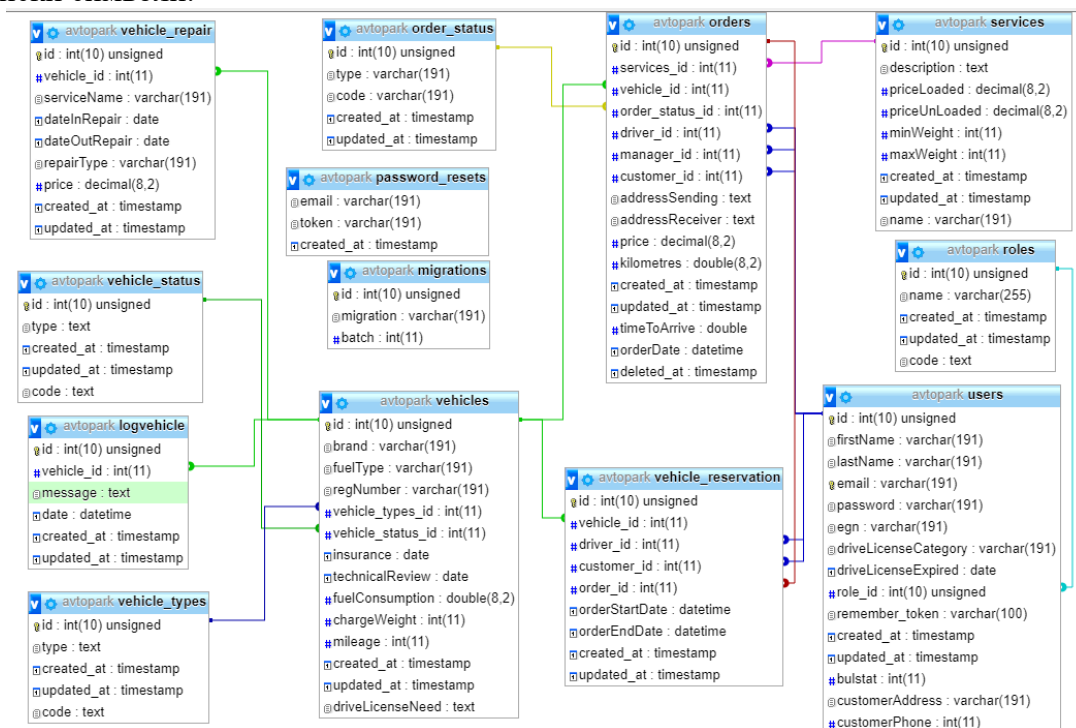


Фиг. 1. Архитектура на Laravel

При разработването на системата е използвана софтуерната рамка Laravel, чиято архитектура е показана на фиг. 1 (*Laravel – The PHP Framework for Web*). От схемата се вижда как е реализиран MVC шаблона за разработка на уеб приложения, а отделните ѝ елементи са: Маршрутизатор (Routing), Контролер (Controller), Модел (Model), Изглед (View), Модул за създаване на заявки (Query Builder) и Миграции (Migration).

2. База данни

Данните в системата се съхраняват в релационна база от данни. Използвана е СУБД (RDBMS) MySQL с колагия *utf8_unicode_ci*, за да могат да се поддържат имена на роли и потребители на кирилица. Освен със символи на кирилица, колагията работи и с други нелатински символи.



Фиг. 2. Релационен модел на базата данни

Релационният модел на базата данни, включващ таблици, техните атрибути и връзки, е показан на фиг. 2. Имената на таблиците подсказват данните, които съдържат.

РЕАЛИЗАЦИЯ НА СИСТЕМАТА

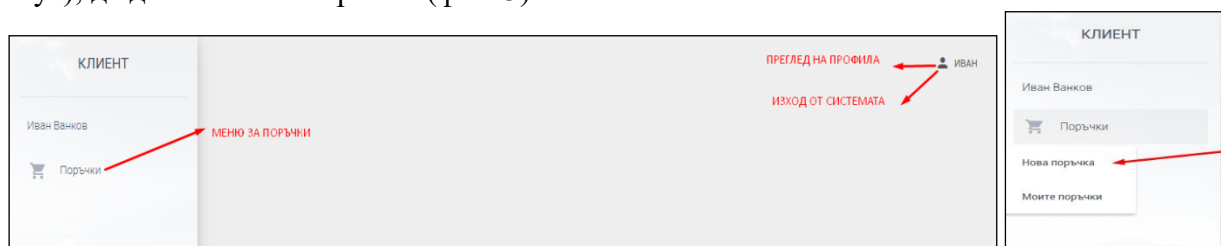
Системата е реализирана в четири основни модула, предназначени за четири категории потребители – Администратор, Мениджър, Клиент и Шофьор. Достъпът до система се получава чрез произволен браузър на адрес <https://avtoparkruse.top>.

1. Модул „Администратор,,

Достъп до административния модул има потребителят администратор, който се създава с инсталирането на системата. Неговата роля се заключава в създаване и управление на останалите категории потребители, включително и други администратори.

2. Модул „Клиент“

От началния екран за оторизация, клиентът може да се регистрира (ако не е) или да влезе в системата със съществуваща регистрация. След като получи достъп до системата може да: преглежда и актуализира профила си; да преглежда състоянието на направени поръчки (техния статус); да добавя нови поръчки (фиг. 3).



Фиг. 3. Основен екран на модул „Клиент“

Основната функционалност на този модул е подаване на поръчка за извършване на транспортна услуга. Това става от избора на опция Поръчки/Нова поръчка (фиг.4).

За всяка поръчка се определят следните задължителни параметри:

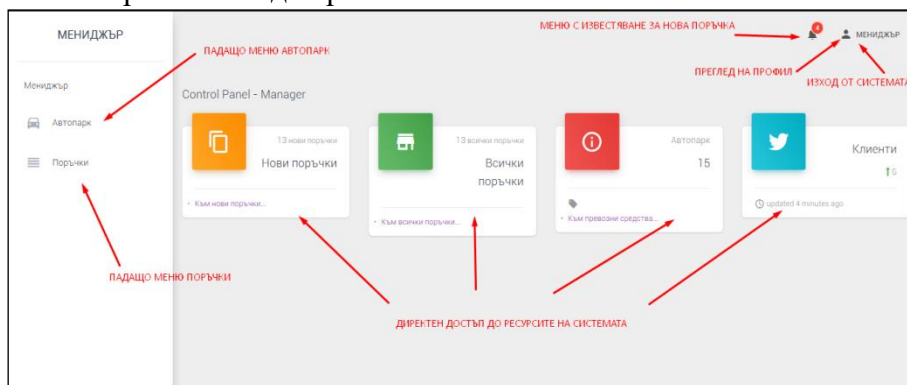
1. Вид на услугата – малки, средни или големи товари, в зависимост от нуждите на клиента.
2. Дата за извършване на услугата – избира се от календар.
3. Маршрут – задава се начален и краен адрес. Това става или чрез картата, или чрез въвеждане ръчно на адресите. Ако ръчно въведеният маршрут е коректен и съществува в Google Maps, разстоянието и времето ще се изчисли автоматично, иначе се въвежда.

Фиг. 4. Добавяне на поръчка

След въвеждане на тази информация (задължителна) се подава поръчката към транспортната фирма от бутон „Поддай поръчка“.

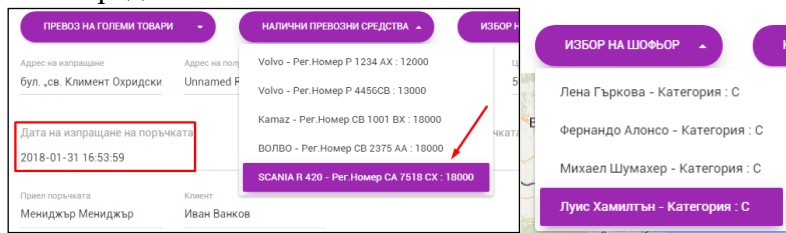
3. Модул „Мениджър“

Този потребител се създава от администратора и основната му роля е да обработва получени заявки за поръчки или да прави нови такива.



Фиг. 5 Основен екран на модул „Мениджър“

В горния десен ъгъл на основния екран на модула се получават известия за новите поръчки (фиг.5). Мениджърът трябва да обработи всяка нова заявка, което включва определяне на конкретно превозно средство и конкретен шофьор, който да извърши услугата (фиг.6). Списъкът с наличните превозни средства се филтрира в зависимост от свободните за датата МПС и вида на товара (големи товари в случая). Аналогично списъкът с шофьори съдържа само тези, които са свободни за определената дата и които имат право да управляват този вид на превозното средство.



Фиг. Избор на превозно средство и шофьор

Мениджърът може и да добавя нова поръчка, като попълва същата информация като клиента и задава МПС и шофьор.

От опцията „Автопарк“ мениджърът управлява наличния автопарк, отстранява и добавя нови превозни средства. За всяко МПС се съхранява информация за – марка, регистрационен номер, тип двигател, вид МПС, статус към момента, изминати километри, максимален товар и дневник с данни за техническата му поддръжка (фиг.6).

Всички превозни средства							
Марка	Номер	Двигател	Тип	Статус	Километри	Товар	Действия
Fiat	78789	dizel	Бус	Свободен	3831	2850	🔍 📄 🗑️
Ford	1245	dizel	Бус	На път	2073	2500	🔍 📄 🗑️
Ford	12355	dizel	Бус	Свободен	5590	2500	🔍 📄 🗑️
Golf2	1111	benz	Кола	В ремонт	1271	850	🔍 📄 🗑️
Kamaz	CB 1001 BX	dizel	Камцион	Резервиран	18996	18000	🔍 📄 🗑️
Lada	12345	benz	Кола	Свободен	8093	1200	🔍 📄 🗑️
SCANIA R 420	CA 7518 CX	dizel	Камцион	Свободен	68294	18000	🔍 📄 🗑️

Фиг. 6. Състояние на автопарка към даден момент

4. Модул „Шофьор“

Този потребител също се създава от администратора. Основната функция на този модул е да информира за нова поръчка, а шофьорът да я изпълни. Когато мениджърът избере конкретен шофьор за изпълнение на услуга, то той автоматично получава известие. Целият

процес по обработването на една заявка за транспортна услуга е представен в следващите екрани:

ID	Услуга	МПС	Начален адрес	Краен адрес	Разстояние	Статус на поръчката	Действия
6	Превоз на големи товари	ШМИТЦ	ul. "Tsanko Tserkovski" 2, 1164 g.k. Lozenets, Sofia, Bulgaria	Славянска 34, 4101 Kuklen, Bulgaria	158	Обработка се	ЗАПОЧВАМ
От 1 до 1 от общо 1 записа					ПЪРВА	ПРЕДИШНА	1 СЛЕДВАЩА ПОСЛЕДНА

Фиг. 7. Шофьорът започва с изпълнението на възложената му поръчка

6	Превоз на големи товари	ul. "Tsanko Tserk Read More	Славянска 34, 410 Read More	158	246.12	Изпратена
ID	Услуга	Начален адрес	Краен адрес	Разстояние	Цена	Статус на поръчката

Фиг. 8. Промяна на статуса на поръчката

6	Превоз на големи товари	ШМИТЦ	ul. "Tsanko Tserkovski" 2, 1164 g.k. Lozenets, Sofia, Bulgaria	Славянска 34, 4101 Kuklen, Bulgaria	158	Изпратена	ПРИКЛЮЧАМ
Статус на							

Фиг. 9. Приключване на поръчката

6	Превоз на големи товари	ul. "Tsanko Tserk Read More	Славянска 34, 410 Read More	158	246.12	Приключена
ID	Услуга	Начален адрес	Краен адрес	Разстояние	Цена	Статус на поръчката
От 1 до 2 от общо 2 записа					ПЪРВА	ПРЕДИШНА 1 СЛЕДВАЩА ПОСЛЕДНА

Фиг.10. Статус на поръчката „Приключена“

Всяко от действията на шофьора (фиг. 7 и фиг. 9) съответства на етап от изпълнението на поръчката. Автоматично се променя и статусът на заявката, който вижда клиента (фиг. 8 и фиг. 10).

ИЗВОДИ

Разработената уеб система представлява напълно завършен продукт за контрол и управление дейността на автопарк. Системата покрива изискванията за качество на съвременна и функционална система от този тип. В бъдеще може да бъде разработен и включен GPS модул и мобилна версия. Тези подобрения ще бъдат изключително полезни за управление на повече параметри от автопарка и подобряване на функционалността на модул „Шофьор“.

REFERENCES

Transportation Management Software - https://www.capterra.com/transportation-management-software/?utf8=%E2%9C%93&users=&sort_options=Highest+Rated (Accessed on 27.03.2018)

Comparison of Transport Management Systems - <https://www.capterra.com/transportation-management-software/compare/137508-131505-156155/DispatchMAX-vs-MyRouteOnline-vs-Transporters-io/> (Accessed on 27.03.2018)

Google Maps API - <https://developers.google.com/maps/> (Accessed on 27.01.2018)

Laravel – The PHP Framework for Web - <https://laravel.com/> (Accessed on 12.01.2018)

THU-1.416-SSS-I-03

METHODOLOGY & TECHNIQUES USED IN THE DEVELOPMENT OF GRAPHICS LIBRARY WITH OPENGL ¹²

Svetlozar Iliev - Bachelor in Computer Science

Department of Informatics & Information technology,

University of Ruse "Angel Kanchev"

Tel.: +359 883 517 829

E-mail: svetlozar.iliev@sirma.bg

Prof. Tsvetomir Vasilev, PhD

Department of Informatics & Information technology,

University of Ruse "Angel Kanchev"

Phone: +359 82 88 84 75

E-mail: tvassilev@uni-ruse.bg

Abstract: This paper describes & reviews the development stages of a graphics library, called by its developer GFX, and also summarizes its main characteristics. This paper also describes the core techniques used in the development stages of such library and all additional libraries required for the completion of the target functionality. It covers all aspects and requirements of a traditional graphics library as well existing solutions and their applications in the industry. The Paper's main target is to explain and lay down the basics of a traditional computer graphics system as well as specifics regarding the actual library implementation. The author of this project was motivated by the potential acquisition of knowledge and experience in this field and also the chance of developing a library which addresses some of the most common shortcomings in present day state of the art graphics libraries.

Keywords: OpenGL, Computer Graphics, Linear Algebra, Rendering & Optimisation techniques.

ВЪВЕДЕНИЕ

Статията има за цел да опише етапите на изграждане на типична графична библиотека подобна на добре познатите библиотеки като Ogre3D, Irrlicht, Horde3D, Unity, Unreal и др., както и да обобщи по важните ѝ характеристики. Идеята за създаване на такава библиотека се корени в няколко причини, част от които са потенциалното придобиване на по-детайлни познания в сферата на компютърната графика и създаване на нетипична, гъвкава библиотека, която решава някои от основните проблеми съществуващи в горе споменатите съществуващи към днешен ден решения.

Произход и предназначение на графичните библиотеки – терминът графична библиотека (на английски – graphics engine) е често срещана част от една по-голяма единица позната в света на примерната графика като „game engine” се заражда още в средата на 90-те години на миналия век във връзка с една от най-известните игри от т.нар. „first person shooter” жанр, наречена „Doom”. Това е една от първите по-популярни игри, познати на пазара, при която ясно се изгражда видима граница или разделение между основното ядро и компонентите на графичната библиотека. Значението на това разделение бързо станало очевидно, когато разработчици започнали да използват отново съществуващи графични библиотеки без никакви или с минимални промени заедно с други компоненти за създаване на напълно нови и различни виртуални игри и графични приложения. Това дало и началото на т.нар. “mod” студия, които създавали нови игри, използвайки и преправяйки съществуващи вече такива.

¹² Докладът е представен на студентската научна сесия на 03. 05. 2018 г. в секция „Информатика“ с оригинално заглавие на български език: ПОДХОД ПРИ РАЗРАБОТВАНЕ НА ГРАФИЧНА БИБЛИОТЕКА С ИЗПОЛЗВАНЕ НА OPENGL

Игри като Quake III Arena & Unreal са типичен пример за виртуални игри създадени с идеята да бъдат лесно променяни и гъвкави, използвайки скриптов езици, които да управляват вътрешната логика на играта, без да има нужда да се променят компонентите, които стоят в основата (част от които е и графичната библиотека) за създаването им.

Структура на графичните библиотеки – в основата си една графична библиотека се състои от няколко основни компонента или слоеве на абстракция, които работят заедно, за да постигнат ефективно, реалистично и оптимално крайната цел – изобразяване на компютърно генерирани изображения.

- *Първи слой* – най-ниското ниво на абстракция в една графична библиотека, това е нивото, което комуникира и работи директно с хардуера или със съществуващи Graphics API като OpenGL, DirectX, Vulkan, които се грижат за комуникацията с графичния драйвер.
- *Втори слой* – междинно ниво, което обвива първия слой на абстракция и служи за премахване или скриване на специфични ограничения и характеристики, съществуващи в използвания API, драйвър или слой от предходното ниво. Това ниво дава възможност библиотеката да стане независима от вида, характеристиките, ограниченията или особеностите на използвания графичен конвейер и хардуер. В това ниво се изграждат основните функционалности, стоящи в основата на графичната библиотека като създаване, изтриване, копиране на видео буфери, текстури, шейдъри (на английски – shaders), графични контексти, интерфейс буфери, графични слоеве, хардуерни заявки и др.
- *Трети слой* – това ниво се характеризира с висока абстракция, в него обикновено са включени функционалности, с които работи крайният потребител или клиент на една графична библиотека. Най-често този слой се характеризира с това, че съдържа много по-голям брой функционалности спрямо слоевете от по-ниско ниво. В този слой се съдържат елементи за визуализация и импортиране на обекти, осветяване, сенки, текстуриране, тримерно селектиране, сериализация, симулация на частици, симулация на течности, премахване на невидими обекти, прозрачност, смяна на проекции, прилагане на филтри, управление на входа и изхода и др.

СЪЩЕСТВУВАЩИ РЕШЕНИЯ

Към днешен ден съществуват голям набор от графични библиотеки. Практически те се разделят на две основни категории. От една страна съществуват библиотеки, които са доста гъвкави и тяхното предимство е, че те предоставят само основните базови графични функционалности, които могат да бъдат лесно разширявани и персонализирани във всяка една посока. Голям недостатък при този вид библиотеки е, че изискват голям набор от познания, за да бъдат използвани или изменени. Друг недостатък е количеството време, което се изисква, за да се изгради минимален, функциониращ проект. Такива библиотеки често изискват от потребителите да имат дълбоки програмни познания, те не предлагат лесен и удобен начин за взаимодействие, като напр. графичен интерфейс, често не включват допълнителни системи като аудио система или система за физична симулация. Няколко примера за такива библиотеки са:

- *Ogre3D* – е базова графична библиотека, която осигурява основните елементи нужни за работата на графичния конвейер. Ogre3D предлага различни методи за визуализация и управление на сцени включващи пространствено разделяне, странициране, портали.

Друга характерна особеност е и възможността за работа с динамично ниво на детайл, анимации, работа с шейдъри на различни езици (GLSL, HLSL, Assembly, Cg)

- *Cocos2d* – е библиотека с отворен достъп, която предлага функционалности за моделиране, визуализиране и симулиране на двумерни графични среди. Известна е с това, че е една от библиотеки с минимален набор функции, но и една от най-бързите към днешна дата. Дава възможност за създаване на проекти за всякакви устройства, както мобилни така и десктоп. Бързината ѝ дава възможност да се разработват приложения за устройства с по-слаб хардуер спрямо съвременните стандарти.
- *Irrlicht* - е библиотека с отворен достъп предлагаща най-голям набор от функционалности спрямо Ogre3D или Cocos2d. Характеризира се с това, че е сравнително малка, което дава възможност за лесно и бързо овладяване, поддържана е от голяма общност и е свободна за голям набор от програмни езици като C++, Java, Lua, Pascal, Basic, Python и др. Irrlicht поддържа OpenGL, DirectX 8, 9, and 11, както и шейдъри за различни езици (GLSL, HLSL). В сравнение със своите съперници тя предлага не само набор от графични функционалности, но и възможност за четене, запис и възпроизвеждане на голям набор от аудио, видео формати. Irrlicht предлага включена в себе си библиотека за физична симулация.

В другия край на спектъра съществуват библиотеки, които включват огромен набор от функционалности, даващ възможност за бърза разработка на проекти във всякакъв мащаб. Предимството при тях е, че се изискват значително по-малко познания в сферата на компютърната графика, с тях обикновено потребителят взаимодейства чрез потребителски графичен интерфейс, често не изискват пряко познание в сферата на програмирането. Друго характерно предимство е, че те често включват не само набор от методи за графична визуализация, но и други елементи като аудио система или система за физична симулация. Недостатък е, че не могат да бъдат лесно персонализирани или изменяни за работа изискваща специфични нужди или техники.

- *Unreal Engine* – една от най-успешните и най-известните библиотеки на разработчиците от Epic Games, влязла в книгата на рекордите на Гинес. Предназначена е за разработка на големи проекти. Характерно за Unreal Engine е, че се поддържа от множество платформи като Windows, Mac, Linux, iOS, Android, Playstation, Xbox и др. Широко използвана от известни фирми за разработка на игри и графични приложения, някои от които Capcom, Activision, Ubisoft, Microsoft Studios, Nintendo и др.
- *Unity* – друга широко използвана библиотека, която е основно използвана и популярна в кръговете за създаване на мобилни графични приложения или игри. Пригодена е за лесно и бързо разработване и използване, както от професионалисти така и от начинаещи разработчици. Поддържа разработване на двумерни и тримерни графични приложения, притежава гъвкава система за моделиране и анимация. Както своя съперник Unreal Engine, Unity е мулти-платформена библиотека разработвана за Windows, Mac, iOS, Android, Playstation, Xbox, Windows Phone, Tizen и др. Някои от най-известните ѝ потребители са Electronic Arts, LEGO, Ubisoft, Square Enix и др.
- *CryEngine* – подобно на вече споменатите гиганти в индустрията CryEngine не отстъпва по наличния набор от функционалности, които предлага. Едно от големите предимства на тази библиотека пред своите конкуренти се дължи на това, че нейните разработчици предлагат пълния ѝ изходен код напълно безплатно. Характеризира се също с своята

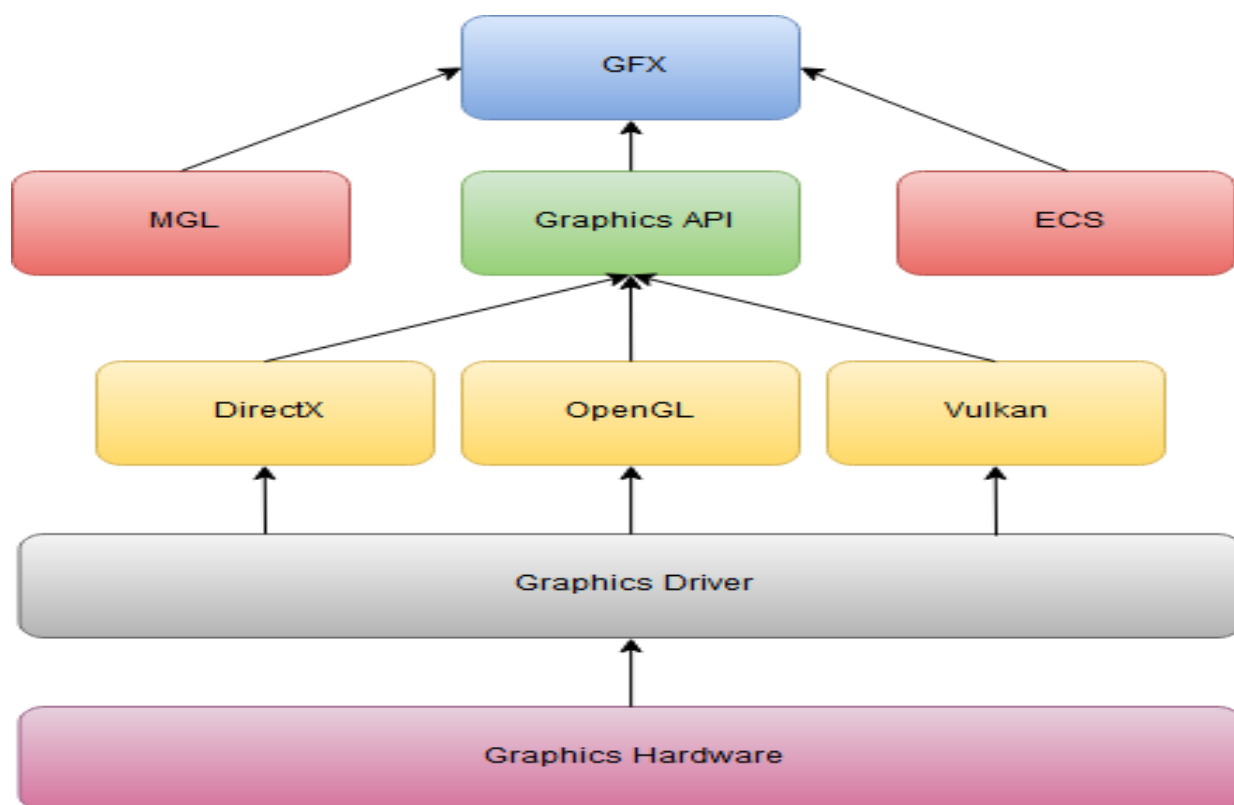
мултиплатформеност за iOS, Android, Windows, Linux, Playstation, Xbox, and Wii. Известни компании като Poppermot Productions, CI Games, Obsidian Entertainment са нейни потребители.

ПРОЕКТИРАНЕ НА ГРАФИЧНА БИБЛИОТЕКА

Цел и задачи – целта на настоящата работа е да се разработи графична библиотека с цел обучение на студенти, която може да послужи и за основа на разработване на цялостна библиотека за симулация в реално време. Друга основна цел на разработката е запознаване със съвременни техники за създаване на компютърно генерирани изображения в реално време, оптимизации и др. Тази библиотека трябва да притежава следните основни възможности:

- Четене на външни файлове съдържащи графични модели или елементи;
- Четене на външни файлове съдържащи графични изображения, текстури, атласи и др.;
- Четене и компилиране и изпълняване на шейдър програми, съдържащи във външни файлове. Зареждане на данни в променливи и шейдър програми;
- Набор от различни възможности за визуализиране на изображения и геометрични елементи в реално време;
- Филтрация и обработка на изображения – изглаждане, осветляване, оразмеряване, подсилване на високи и ниски честоти, регулация на контраста и др.;
- Работа с динамично геометрично ниво на детайл за графични модели
- Поддръжка на основни техники за изобразяване – осветление, текстуриране, сенки, рендиране, изключване на невидими обекти;
- Поддръжка на напреднали техники за изобразяване и симулация на океан, атмосфера, облаци, околна среда и др.;
- Динамично взаимодействие в реално време със виртуалната сцена – селектиране, трансформиране и обработка на обекти;
- Разпознаване на сигнали и команди от набор периферни устройства – мишка, клавиатура, джойстик, трекбол и др.;
- Сериализация и десериализация на виртуална сцена върху външен носител или постоянна памет.

Проектиране - разработването на графичната библиотека представлява набор от три самостоятелни библиотеки, които са разработени и предназначени да работят заедно за реализация на основните цели и задачи на проекта. Предвидено е библиотеката да развива собствени интерфейси (API) и собствен дизайн на модулите, без да се базира върху съществуващи библиотеки, доколкото това е възможно. Фигура 1 систематизира структурата на библиотеката и модулите работещи заедно за реализиране на поставената цел. Следва детайлно описание и запознаване с трите основни модула на графичната библиотека.



Фиг. 1. Структурна схема на графична библиотека GFX

- **MGL (Math for Graphics)** – е библиотека, създадена за работа с математични действия. Тя е една от трите основополагащи библиотеки, която има за цел да даде възможност лесна за работа с основни математически конструкции. Core е основният модул, съдържащ базовите конструкции като: вектори, матрици, кватърниани, докато модула Struct съдържа голям набор от тримерни примитиви и техните двумерни еквиваленти (сфера, конус, паралелепипед, куб, лъч, равнини, пресечен паралелепипед и др.), както и структури за разделяне на пространството (на английски – spacial partitioning). Библиотеката поддържа всички често използвани методи за реализация на основни математически операции (векторно и скалярно произведение, трансформация на матрици, ротация и интерполация между вектори и кватърниани, нормализация, транспониране, детерминанта и намиране на обратна матрица и др.), разположени в модул Functions, както и голям набор от методи за проверка на пресичане между основните геометрични примитиви, като сфера, паралелепипед, равнина, лъч, пресечена пирамида и др., разположени в модул Geometric Test. Важен компонент от тази библиотека са структурите за „ускорение“ на управлението и обработката на геометрични примитиви. Библиотеката поддържа две такива структури наречени: Осмично дърво (на английски – octree) и Точково дърво (на английски – point cloud tree), разположени в модул Space Partition. Осмичното дърво е специален вид дърво, при който всеки родителски възел има осем наследника. Всеки възел представлява куб с размери степени на 2 (4, 8, 16, 32 и т.н) и се дели на 8 идентични куба, които стават негови наследници. Всеки възел съхранява своите наследници и набор от вписани в него примитиви, този вид структури са често използвани за отстраняване на невидими обекти от динамични сцени. Друг важен модул който предлага MGL е API модула, предоставящ набор от програмни интерфейси.

- **ECS (Entity Component System)** – е втората основна библиотека, предоставяща имплементация на архитектурен шаблон често използван при разработката на виртуални игри и графични приложения. Този шаблон е характерен с това, че набляга на използването на композиция пред наследяване, което предоставя голяма гъвкавост. Той се състои от три елемента наречени Обект, Компонент и Система. Компонент в този шаблон представлява проста минимална структура, в която се съхраняват само данни (пример: компонент описващ позицията на един обект би представлявал структура съдържаща само един член - тримерен вектор описващ позицията на обекта в пространството) всеки компонент има уникален идентификатор или номер и описва определена характеристика или действие. Всеки Обект представлява уникален идентификатор или маска, която се получава на база на това какви компоненти са обвързани с него. В зависимост от това с какви компоненти е обвързан даден обект, той притежава различни свойства, възможности и характеристики. Това, което контролира, управлява и използва Компонентите на един Обект са Системите, те са функции или класове описващи набор от действия които обработват определени Компоненти принадлежащи на всеки един от Обектите.
- **GFX (Graphics Effects)** – е третия последен и основен елемент, който комбинира предходните два. Той представлява ядрото на графичната библиотека заедно с пълния набор от функционалности нужни за изпълняване на задачата и целта на проекта. Тук са обособени няколко критични модула, нужни за работата на библиотеката. Първият от тях е модулът, съдържащ всички Компоненти описващи и дефиниращи държанието и характера на Обектите. Друг характерен модул е модулът, описващ структурата и начина на представяне на графичните модели, нужен за зареждане и интерпретиране на външни ресурси, съдържащи геометрични форми, елементи или модели (на английски – mesh). В Основния модул се съдържат набора от Системи изпълняващи действията нужни за пълното изпълняване на вече дефинираната цел на проекта. Групата от системи тук се дели в няколко основни категории. Основни (на английски – Core) - набор от основни системи, които се заемат да решат базови задачи като трансформации, композиране, селектиране, управление на ниво на детайлност, премахване на невидими обекти и др. Друга група са системите отговорни за осветяване и осветление, както и групата от системи, отговорни за генериране на динамични сенки в реално време. Следваща е групата системи за контрол и симулация на частици, вода, океан, облаци и климат – околна среда.

ИЗВОДИ

Създаването на графична библиотека с образователна цел е реализируема задача, дори в рамките на няколко месеца от сам човек или група хора. Напълно възможно е и развиването на библиотеката до състояние, което е готова за използване за практически или реални задачи.

За да има смисъл развиването на даден продукт, отвъд изследователска цел, е необходимо откриване на подходяща ниша, към която продуктът да бъде насочен. На този етап може да се твърди, че нишата за графични системи и библиотеки е успешно заета от наложилите се продукти. Голям набор от тези библиотеки разполагат с огромен брой хора, работещи над тях, постоянно добавяйки нови функционалности, разширявайки ги в разнообразни посоки, включващи не само графичната визуализация.

Като бъдеща посока за развитие може да се продължи с изследване и експериментиране с различни концепции и техники за графично визуализиране. Работа с анимации, техники за реалистично текстуриране, динамично теселиране на обекти в реално време и др.

REFERENCES

- [1] Fletcher Dunn, Ian Parberry - *3D Math Primer for Graphics* 2nd Edition
- [2] Jason Gregory - *Game Engine Architecture*
- [3] Thomas Akenine, Eric Haines, Naty Hoffman - *Real-Time Rendering* 3rd Edition
- [4] David Wolff - *GLSL CookBook* 4.0
- [5] Steve Shreiner, Graham Sellers, John Kessenich - *OpenGL Programming Guide* 8th Edition
- [6] Eric Lengyel – *Mathematics for 3D Computer Graphics* 3rd Edition
- [7] Scott Meyers - *Effective STL*
- [8] Scott Meyers – *Effective C++* 3rd Edition
- [9] Davide Di Gennaro – *Advanced Template Metaprogramming in Classic C++*

THU-1.416-SSS-I-04

BYPASSING ANTIVIRUSES IN 2018¹³

Svetoslav Hadziivanov – Student

Department of Informatics and Information Technologies,

“Angel Kanchev” University of Ruse

Phone: +359884015715

E-mail: warrolen@gmail.com

Abstract: Many popular Antivirus (AV) solutions claim to be effective against unknown malware which is wrong. This paper presents various techniques used by malware developers to evade detections as like as a protection from it. It comes that some AV solutions have better methods to detect x86 malware than others but when it comes to x64 malware, they are all way behind the truth.

Keywords: malware, evade detections, protection

INTRODUCTION

The AV solutions can hardly compare each other. Generally, there are 3 stages in the detection process: Scantime detection (Static detection), Code Emulation (before execution) and Runtime detection (during execution). The first stage is used by all popular AV vendors for a long time. It's based on signatures from known malware files. The AVs are analyzing every file's icon, assembly information, Portable Executable [1] (PE) headers, sections, etc. If there is anything wrong with any of these, the file entropy coefficient increases and when it reaches a certain value, the file is being detected. Any executable can be inspected by using a PE Viewer, for example – CFF Explorer. It is an open source which is completely free. In fig. 1 and fig. 2, a part of the analyzed structures is presented for each scanned file on <http://virustotal.com>.

Portable Executable Info ⓘ

Header

Target Machine	Intel 386 or later processors and compatible processors
Compilation Timestamp	2013-07-29 13:53:11
Entry Point	63550
Contained Sections	4

Sections

Name	Virtual Address	Virtual Size	Raw Size	Entropy	MD5
.text	8192	55364	55808	5.7	a11fa451ebcab3a2fd166a34cec44e5
.sdata	65536	189	512	2.63	e19829aa0459a4f8fc65191000ed9ea7
.rsrc	73728	69248	69632	3.48	f14f084a81da6ad777d6d936dc0c5034
.reloc	147456	12	512	0.08	391becfb5cb3c92ca849dca0e02d025c

Imports

mscoree.dll

Fig. 1 <http://virustotal.com>

¹³ Докладът е представен на студентската научна сесия на 03. 05. 2018 г. в секция „Информатика“ с оригинално заглавие на български език: БАЙПАСВАНЕ НА АНТИВИРУСНИТЕ ПРЕЗ 2018 ГОД.

Contained Resources By Type			
RT VERSION	1		
RT MANIFEST	1		
RT ICON	1		
RT GROUP ICON	1		
Contained Resources By Language			
NEUTRAL	4		
Contained Resources			
SHA-256	File Type	Type	Language
55e86c2fd9084587181adf9ea05270a4242ee82042ca4509fc590c5dad2de382	data	RT_ICON	NEUTRAL
12598188b44d76a8828aa7a8211c4c1bfa8093f617928f5c8f3da9cd81a42d64	ASCII text	RT_GROUP_ICON	NEUTRAL
031626eb3cc9981626b96f4fb74d21aad652f46af021aae86fde0a79ba1d02	data	RT_VERSION	NEUTRAL
c0085eb467d2fc9cf395047e057183b3cd1503a4087d0db565161c13527a76f	data	RT_MANIFEST	NEUTRAL

Fig. 2 <http://virustotal.com>

The first phase is easily passed by obfuscating/packing the file (making the file less readable). Although, there are non-malicious files that often get detected by a mistake. Through the second phase, the AV is running executables in an emulated environment which is monitoring for suspicious behavior. At the last phase, some AVs were found to implement dynamic techniques to identify suspicious behavior while the malware is executing on the system.

PORTABLE EXECUTABLE OVERVIEW

Windows PE Loader requires a specific format so the executables can run. Since the AVs also know that obfuscators/packers are often used to evade static detections, they also attempt to detect anomalies in PE's file format. The executables contain two high-level parts: header and sections. The header contains metadata about the executable and it describes the sections. The sections are collections of bytes put in a certain order by the compiler.

Executable Header

The executable header consists of the following structures as shown below (fig. 3):

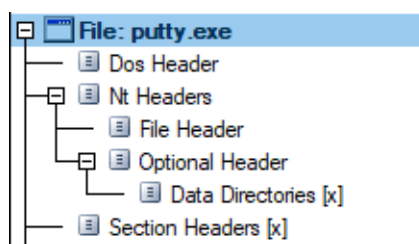


Fig. 3 Executable Header's structure

DOS Header

The DOS header is remaining from the old DOS MZ executable format. The first two bytes of the DOS header correspond to e_magic field that must be equal to 0x54AD – „MZ”. The default PE DOS stub should contain „This program cannot be run in DOS mode” text which is ignored by the modern Windows Operating Systems (OS) and they only verify the „MZ” and the offset to the PE header. If the file doesn't contain that, the file will be flagged as suspicious. This is shown in fig. 4.

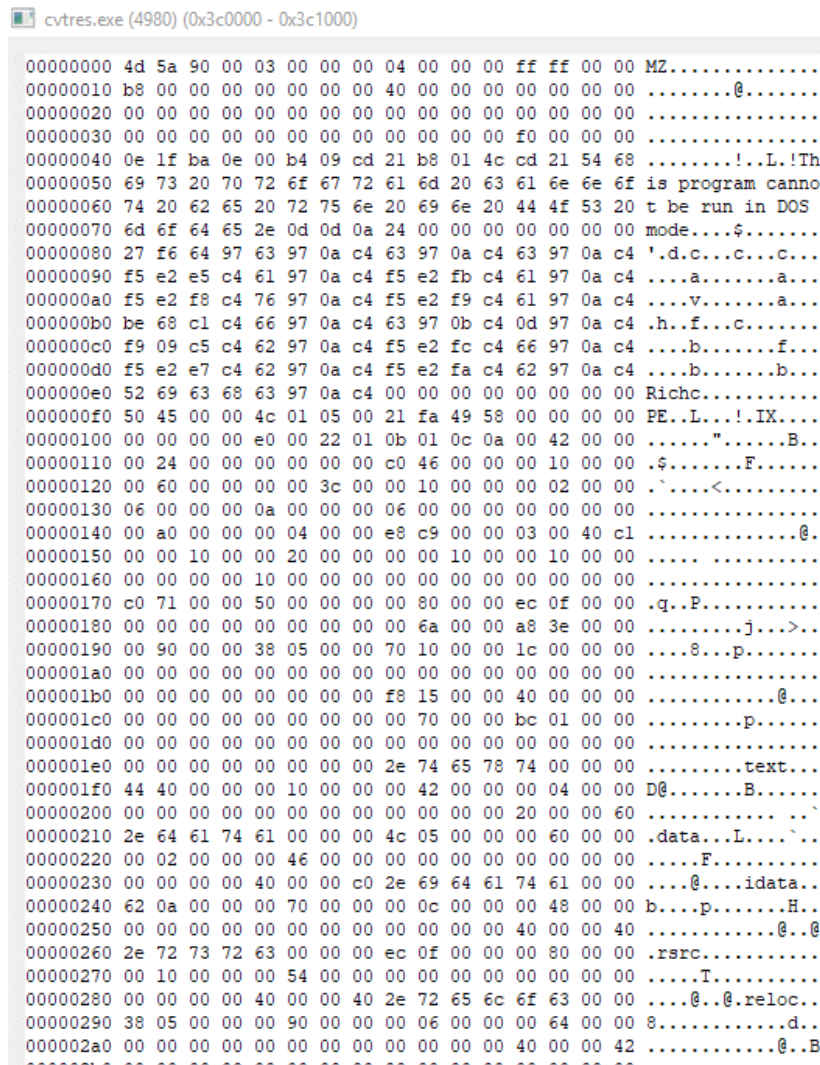


Fig. 4 PE in memory

NT Header (often referred as PE header)

It starts with a static signature value equal to 0x4550 – „PE”. It is also verified by Windows PE Loader before loading the executable in memory. It consists of two subsections: File Header and Optional Header.

- **File Header**

The only fields of that structure that were found to be checked by the AV vendors are: TimeDateStamp, PointerToSymbolTable and NumberOfSymbols. If the TimeDateStamp has unrealistic value or the rest two fields aren't zeros, the file entropy value increases. It is shown in fig. 5:

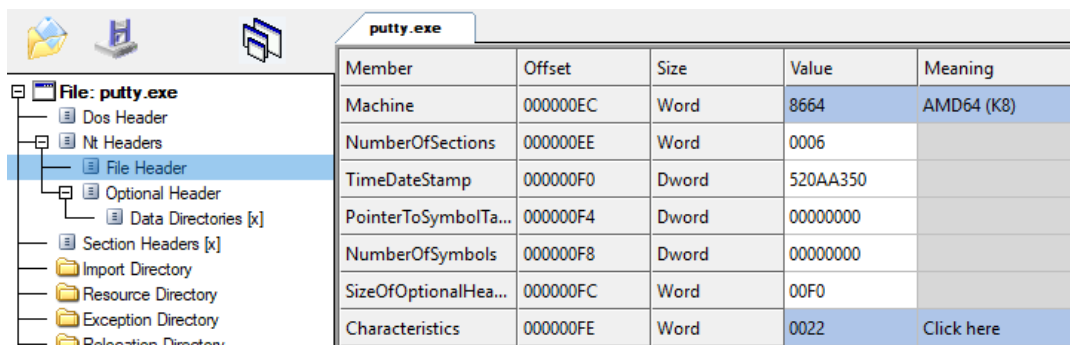


Fig. 5 File Header fields

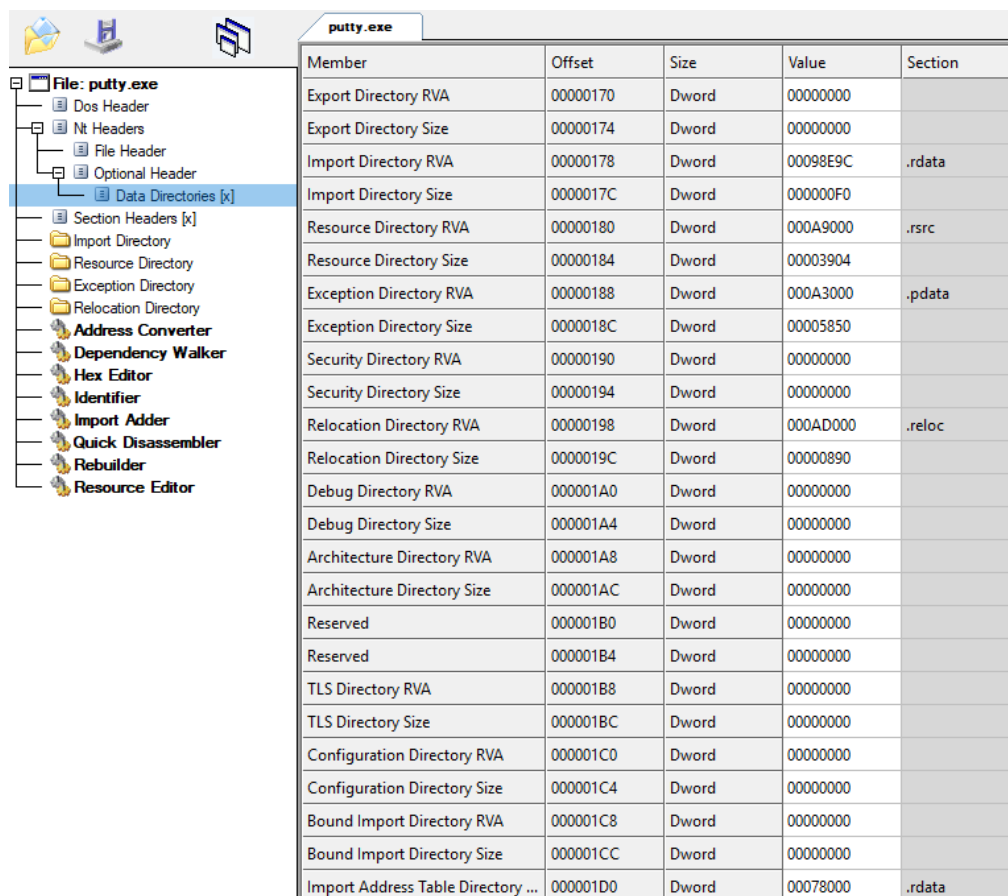
Member	Offset	Size	Value	Meaning
Magic	00000100	Word	020B	PE64
MajorLinkerVersion	00000102	Byte	0A	
MinorLinkerVersion	00000103	Byte	00	
SizeOfCode	00000104	Dword	00076400	
SizeOfInitializedData	00000108	Dword	0002F600	
SizeOfUninitializedData	0000010C	Dword	00000000	
AddressOfEntryPoint	00000110	Dword	000691B8	.text
BaseOfCode	00000114	Dword	00001000	
ImageBase	00000118	Qword	0000000140000000	
SectionAlignment	00000120	Dword	00001000	
FileAlignment	00000124	Dword	00000200	
MajorOperatingSystemVers...	00000128	Word	0005	
MinorOperatingSystemVers...	0000012A	Word	0002	
MajorImageVersion	0000012C	Word	0000	
MinorImageVersion	0000012E	Word	0000	
MajorSubsystemVersion	00000130	Word	0005	
MinorSubsystemVersion	00000132	Word	0002	
Win32VersionValue	00000134	Dword	00000000	
SizeOfImage	00000138	Dword	000AE000	
SizeOfHeaders	0000013C	Dword	00000400	
Checksum	00000140	Dword	000B09F8	
Subsystem	00000144	Word	0002	Windows GUI
DllCharacteristics	00000146	Word	8140	Click here
SizeOfStackReserve	00000148	Qword	0000000000100000	
SizeOfStackCommit	00000150	Qword	0000000000001000	

Fig. 6 Optional Header fields

- Optional Header

The Optional Header contains a lot of information from the Section Headers (fig. 6).

The SizeOfCode, SizeOfInitializedData, SizeOfUninitializedData, BaseOfCode and BaseOfData fields are directly related to information found in the Section Headers at the very end of the executable's header. If there is any inconsistency between them, several AV solutions raise detections. The most important fields in this section are: SizeOfHeaders, SizeOfImage and AddressOfEntryPoint. They are used when we want to inject an executable in memory. The AddressOfEntryPoint contains the offset to the entry point where the PE loader will start executing from. It is often modified in order to hijack execution which also flags a detection.



Member	Offset	Size	Value	Section
Export Directory RVA	00000170	Dword	00000000	
Export Directory Size	00000174	Dword	00000000	
Import Directory RVA	00000178	Dword	00098E9C	.rdata
Import Directory Size	0000017C	Dword	000000F0	
Resource Directory RVA	00000180	Dword	000A9000	.rsrc
Resource Directory Size	00000184	Dword	00003904	
Exception Directory RVA	00000188	Dword	000A3000	.pdata
Exception Directory Size	0000018C	Dword	00005850	
Security Directory RVA	00000190	Dword	00000000	
Security Directory Size	00000194	Dword	00000000	
Relocation Directory RVA	00000198	Dword	000AD000	.reloc
Relocation Directory Size	0000019C	Dword	00000890	
Debug Directory RVA	000001A0	Dword	00000000	
Debug Directory Size	000001A4	Dword	00000000	
Architecture Directory RVA	000001A8	Dword	00000000	
Architecture Directory Size	000001AC	Dword	00000000	
Reserved	000001B0	Dword	00000000	
Reserved	000001B4	Dword	00000000	
TLS Directory RVA	000001B8	Dword	00000000	
TLS Directory Size	000001BC	Dword	00000000	
Configuration Directory RVA	000001C0	Dword	00000000	
Configuration Directory Size	000001C4	Dword	00000000	
Bound Import Directory RVA	000001C8	Dword	00000000	
Bound Import Directory Size	000001CC	Dword	00000000	
Import Address Table Directory ...	000001D0	Dword	00078000	.rdata

Fig. 7 Data Directories fields

- Data Directories

The data directories are a subsection of Optional Header. This is a table of offsets to the specific directories in an executable. Import directory and relocations directory must be resolved in order to run specific executable in memory. The IAT (Import Address Table) references a structure that lists the functions of external DLLs on which the executable relies (for example: Kernel32.dll, NTDLL.DLL, etc.). The executable will have relocations directory only if ASLR (Address space layout randomization) is enabled while compiling the file. It contains all global variable addresses and whenever the executable is re-opened, it puts those variables on random places (offsets) in memory. The most Delphi applications require a TLS (Thread-Local Storage [2]) directory which should be resolved as well. The TLS callbacks are executed before the entry point (Main function). There are static TLS slots that should be resolved for each new thread. Data Directories fields are shown in fig. 7.

Section Headers

In fig. 8, the section headers is shown for „putty”. It has 6 sections: „.text” section contains the program’s code; „.rdata” contains specific metadata (in our case: Import Directory, Import Address Table Directory); „.data” – static data from the code (e.g. hard-coded static strings); „.pdata” – Exception Directory; „.rsrc” contains additional resources like version information and bitmap icons; „.reloc” has the Relocation Directory which means that ASLR for the executable is enabled. Characteristics field is also important because it contains the protection flags of each sections (read, write and execute), as well as information about the content of the section which is a duplicate from Optional Header. If all sections are RWX (read, write and execute), the AVs are raising suspicion.

Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations N...	Linenumbers ...	Characteristics
Byte[8]	Dword	Dword	Dword	Dword	Dword	Dword	Word	Word	Dword
.text	00076281	00001000	00076400	00000400	00000000	00000000	0000	0000	60000020
.rdata	00022F22	00078000	00023000	00076800	00000000	00000000	0000	0000	40000040
.data	000077C4	0009B000	00002200	00099800	00000000	00000000	0000	0000	C0000040
.pdata	00005850	000A3000	00005A00	0009BA00	00000000	00000000	0000	0000	40000040
.rsrc	00003904	000A9000	00003A00	000A1400	00000000	00000000	0000	0000	40000040
.reloc	00000F70	000AD000	00001000	000A4E00	00000000	00000000	0000	0000	42000040

Fig. 8 Section Headers Overview

```

1 // Stub.Program
2 // Token: 0x06000001 RID: 1 RVA: 0x00002080 File Offset: 0x00002080
3 [STAThread]
4 private static void Main()
5 {
6     Program.BypassAvastSandBox();
7     byte[] array = Program.ReadPayload();
8     string text = string.Empty;
9     bool flag = HelpingMethods.IsDotNet(array);
10    if (flag)
11    {
12        text = Path.Combine(RuntimeEnvironment.GetRuntimeDirectory(), "RegAsm.exe");
13    }
14    else
15    {
16        text = Path.Combine(RuntimeEnvironment.GetRuntimeDirectory(), "AppLaunch.exe");
17    }
18    switch (Convert.ToInt32("[TECHNIQUETYPE]"))
19    {
20        case 0:
21        {
22            ITechnique technique = new RunPE();
23            technique.Run(new object[]
24            {
25                text,
26                array
27            });
28            break;
29        }
30        case 1:
31        {
32            ITechnique technique = new Doppelganger();
33            technique.Run(new object[]
34            {
35                text,
36                array
37            });
38            break;
39        }
40    }
41 }

```

Fig. 9 Stub content (unobfuscated)

```

1 using System;
2 using System.IO;
3 using System.Reflection;
4
5 namespace Loader
6 {
7     // Token: 0x02000002 RID: 2
8     internal class Program
9     {
10        // Token: 0x06000001 RID: 1 RVA: 0x0000205C File Offset: 0x0000205C
11        [STAThread]
12        private static void Main()
13        {
14            byte[] rawAssembly = Program.ReadStub();
15            AppDomain.CurrentDomain.Load(rawAssembly).EntryPoint.Invoke(null, null);
16        }
17
18        // Token: 0x06000002 RID: 2 RVA: 0x00002088 File Offset: 0x00002088
19        private static byte[] ReadStub()
20        {
21            Assembly executingAssembly = Assembly.GetExecutingAssembly();
22            string name = "stub";
23            byte[] result;
24            using (Stream manifestResourceStream = executingAssembly.GetManifestResourceStream(name))
25            {
26                using (new StreamReader(manifestResourceStream))
27                {
28                    byte[] array = new byte[manifestResourceStream.Length];
29                    manifestResourceStream.Read(array, 0, array.Length);
30                    result = array;
31                }
32            }
33            return result;
34        }
35    }
36 }

```

Fig. 10 Loader content (unobfuscated)

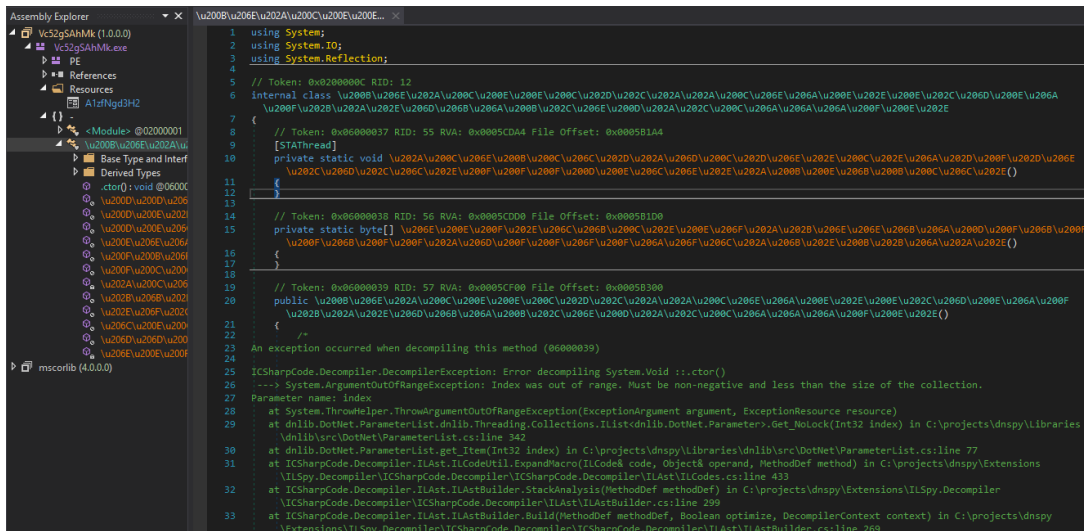


Fig. 11 Loader content (obfuscated)

STUB DESIGN

The most malicious files are composed of a loader and a stub. The loader has the stub bytes encrypted and saved in Resources (or for example in: section headers, byte array, string variable, etc.), it decrypts those bytes at runtime (during execution) and it loads them in memory. The stub's construction is exactly like loader's but instead of the stub bytes it has payload's bytes (the real malware) stored in the Resources and the method of executing those bytes in memory is different than loader's because it would get killed at runtime by some AVs.



Fig. 12 Non-obfuscated/packed malware



Fig. 13 Obfuscated malware

For the research, a privately made example crypter by the author. The most of the malware files are coded in C/ASM so they won't have a dependency and they can run on a fresh installed Windows OS from XP to 10. These examples were made in C# which requires .NET Framework to be installed on targetting system. By default, Windows 8, 8.1 and 10 has .NET Framework 4.0, but Windows XP, Vista and 7 have only .NET Framework 2.0. The examples are using dnSpy for source code view. In fig. 9, 10 and 11, an example stub and loader are presented.

To test whether the obfuscated file passes phase 1 (scantime) or not, the file can be scanned on <http://nodistribute.com>, <http://viruscheckmate.com> or by using installed on PC AVs. Note that <http://nodistribute.com>, <http://virustotal.com>, <https://www.hybrid-analysis.com> and a few others are distributing files to the AV labs for detailed analysis. So, it's not the best idea to scan there while doing a research like this.

Fig. 12 and 13 are showing the differences before obfuscating and after obfuscating a malicious file. Fig. 13 shows that there are 0/37 detections which means that the file is completely FUD (Full Undetectable) at scantime. The analyzed file is 32-bit architecture which matters. The most AVs doesn't have implemented same engines for 64-bit files as like as they do have for 32-bit files. The reason is because the most of the malicious files are made to target 32-bit systems. Anyway, scantime is the beginning of bypassing. For phase 2 (Code Emulation) and phase 3 (Dynamic techniques to detect malware at runtime), there should be some code to prevent each AV strong feature so we will be able to bypass them.

The following AVs were object to the research for phase 2 and phase 3:

- Windows Defender (on Windows 10 which is not trash as like as Windows 7's is)
- AVAST/AVG (Same engine)

- Avira
- ESET NOD32 Antivirus
- BitDefender Antivirus
- Kaspersky Antivirus
- Norton Security
- Malwarebytes Anti-Malware
- 360 Total Security

Not going to discuss AVs like McAfee because they are years behind the above mentioned.

Bypassing Emulation (phase 2)

A few AVs were found to implement emulation technique. A code from the most used anti-emulation technique by malware developers is presented below. The following check uses two timers to detect any sleep emulation. This is not the best way to do this but it is the most used.

```
private static void AntiEmulation()
{
    DateTime then = DateTime.Now;
    int timeout = new Random(Guid.NewGuid().GetHashCode()).Next(800, 1200);
    typeof(Thread).GetMethod("Sleep", BindingFlags.Public | BindingFlags.Static,
null, new Type[] { typeof(int) }, null).Invoke(null, new object[] { timeout });
    DateTime now = DateTime.Now;

    if (((now - then).TotalMilliseconds) < timeout)
    {
        GC.Collect(0, GCCollectionMode.Forced);
        GC.WaitForFullGCComplete();
        GC.WaitForPendingFinalizers();
        throw new BadImageFormatException(null, default(Exception));
    }
}
```

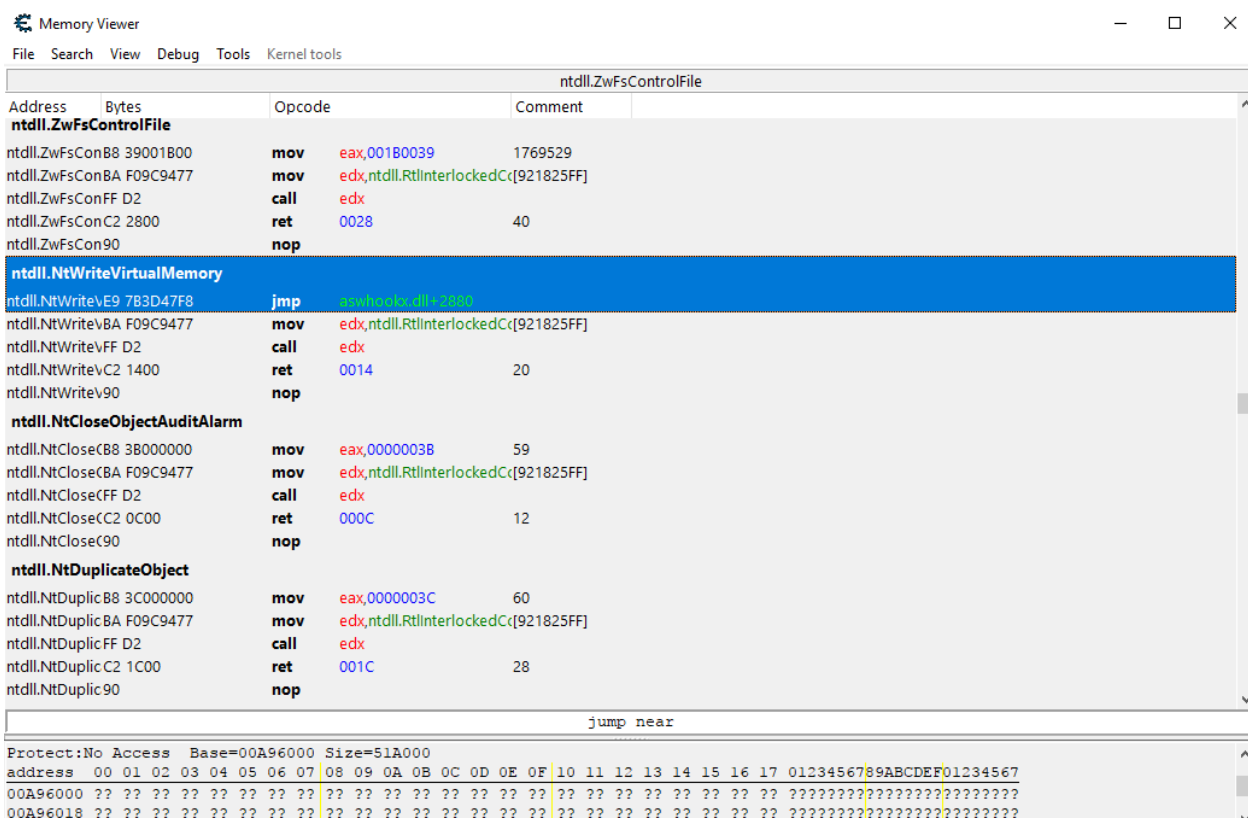


Fig. 14 AVAST/AVG NtWriteVirtualMemory Inline Hook

Bypassing AVs at runtime (phase 3)

- Hooks

AVAST/AVG has implemented Ring3 (userland) inline hooks. They are hooking a few WinAPIs from ntdll.dll which means that kernel32.dll APIs are also affected by it because they refer to ntdll ones at some point. The inline hooks are replacing the very first 5 bytes with a jmp instruction which jumps to a trampoline function used to find suspicious activity. In fig. 14, a hook on NtWriteVirtualMemory is installed. There is also shown that there are a few more API's whose first 5 bytes are normal (not hooked). This can be easily bypassed by using syscalls (C/C++) or manually mapping our own copy of ntdll in the remote process and using the WinAPIs from it (the same way game hacks are bypassing anti cheats).

On the following code, all hooked WinAPIs by AVAST/AVG are presented:

```
[4392] output.exe!ntdll.dll->RtlQueryEnvironmentVariable || [0x7790B6B0] => aswhookx.dll
[0x6FDA35DB] || Inline - Detour [5 Bytes] || push 20h || jmp 6FDA35E0h
[4392] output.exe!ntdll.dll->NtQueryInformationProcess || [0x7792E8D0] => aswhookx.dll
[0x6FDA366B] || Inline - Detour [5 Bytes] || mov eax , 00000019h || jmp 6FDA3670h
[4392] output.exe!ntdll.dll->NtMapViewOfSection || [0x7792E9E0] => aswhookx.dll
[0x6FDA29EB] || Inline - Detour [5 Bytes] || mov eax , 00000028h || jmp 6FDA29F0h
[4392] output.exe!ntdll.dll->NtWriteVirtualMemory || [0x7792EB00] => aswhookx.dll
[0x6FDA287B] || Inline - Detour [5 Bytes] || mov eax , 0000003Ah || jmp 6FDA2880h
[4392] output.exe!ntdll.dll->NtOpenEvent || [0x7792EB60] => aswhookx.dll
[0x6FDA2DEB] || Inline - Detour [5 Bytes] || mov eax , 00000040h || jmp 6FDA2DF0h
[4392] output.exe!ntdll.dll->NtCreateEvent || [0x7792EBE0] => aswhookx.dll
[0x6FDA2C9B] || Inline - Detour [5 Bytes] || mov eax , 00000048h || jmp 6FDA2CA0h
[4392] output.exe!ntdll.dll->NtResumeThread || [0x7792EC80] => aswhookx.dll
[0x6FDA2B8B] || Inline - Detour [5 Bytes] || mov eax , 00070052h || jmp 6FDA2B90h
[4392] output.exe!ntdll.dll->NtCreateMutant || [0x7792F230] => aswhookx.dll
[0x6FDA2F2B] || Inline - Detour [5 Bytes] || mov eax , 000000ADh || jmp 6FDA2F30h
[4392] output.exe!ntdll.dll->NtCreateSemaphore || [0x7792F2E0] => aswhookx.dll
[0x6FDA31BB] || Inline - Detour [5 Bytes] || mov eax , 000000B8h || jmp 6FDA31C0h
[4392] output.exe!ntdll.dll->NtCreateUserProcess || [0x7792F370] => aswhookx.dll
[0x6FDA344B] || Inline - Detour [5 Bytes] || mov eax , 000000C1h || jmp 6FDA3450h
[4392] output.exe!ntdll.dll->NtOpenMutant || [0x7792F910] => aswhookx.dll
[0x6FDA307B] || Inline - Detour [5 Bytes] || mov eax , 0000011Bh || jmp 6FDA3080h
[4392] output.exe!ntdll.dll->NtOpenSemaphore || [0x7792F980] => aswhookx.dll
[0x6FDA330B] || Inline - Detour [5 Bytes] || mov eax , 00000122h || jmp 6FDA3310h
[4392] output.exe!ntdll.dll->RtlDecompressBuffer || [0x77990EB0] => aswhookx.dll
[0x6FDA34FB] || Inline - Detour [5 Bytes] || mov edi , edi || jmp 6FDA3500h
```

Kaspersky and Norton have implemented SSDT hooks (this time – kernel mode). This means that the bypass methods from AVAST/AVG will not work here because our file has only user permissions. Anyway, It doesn't mean that it cannot be bypassed. It can be done by preventing kernel drivers (for the SSDT hooks) from being loaded.

Cloud Protection

Avira and Norton have implement a Cloud Protection which sends the file to their labs for a review. It only happens if the injection method is detected like non-modified Process Hollowing/Dynamic Forking. It can also be bypassed by pumping the file with 3 MBs. This means that if our file is 358 KB, it should be pumped with junk bytes up to at least 3 MBs.

Memory scan

ESET's Memory Scanner (HIPS) is scanning all memory regions in each executable for anomalies/execution patterns of known malware. It can be bypassed by two ways. The first one is to set proper section protections by getting the values from Characteristics field and not only

RWX (read, write and execute) permissions everywhere and making our newly allocated memory for the injected image (payload) either Image or Mapped (fig. 15). The second one is to spam PAGE_GUARD (protects memory region) on the headers which doesn't break the performance in any way.

Address	Type	Size	Permissions	Notes
> 0x8d0000	Mapped	8 kB	R	
> 0x8f0000	Private	64 kB	RW	Heap 32-bit (ID 3)
> 0x900000	Mapped	8 kB	R	
> 0x960000	Private	1,024 kB	RW	Heap 32-bit (ID 1)
> 0xa70000	Image	780 kB	WCX	
0xa70000	Image: Commit	156 kB	R	
0xa97000	Image: Commit	20 kB	RW	
0xa9c000	Image: Commit	4 kB	R	
0xa9d000	Image: Commit	12 kB	WC	
0xaa0000	Image: Commit	544 kB	RX	
0xb28000	Image: Commit	44 kB	R	
> 0xb40000	Private	1,024 kB	RW	Stack 32-bit (thread 5044)
> 0xc40000	Private	1,024 kB	RW	Stack 32-bit (thread 6420)

cvtres.exe (9260) (0xa70000 - 0xa97000)

```

00000000 4d 5a 00 00 00 00 00 00 00 00 00 00 00 00 00 00 MZ.....
00000010 00 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 .....8.....
00000020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000030 00 00 00 00 00 00 00 00 00 00 00 00 40 00 00 00 .....8...

```

Fig. 15 Proper memory protections

INJECTION TECHNIQUES

Process Hollowing/Dynamic Forking (RunPE)

Process Hollowing/Dynamic Forking is the most used technique by malware developers. It spawns a suspended legit child process (like svchost.exe), allocates space inside it for the malware, copies PE headers and sections to that space, sets proper protection flags to the sections, patches Process Environment Block (PEB) [3] (modifying the base address to the new one) and then releases the thread. The AV solutions could have easily implemented a method which scans for mismatching fields from the PE on disk and PE in memory. Therefore, they would have killed this method long time ago but it looks like they don't care or they have no idea of what they are facing.

Process Doppelgänger

Process Doppelgänger is a new technique which uses NTFS cache system. It creates a transaction and then a transacted file on the disk which is not visible though because of the NTFS's cache. It then writes payload bytes into it and it creates a new section which we later map into the chosen process.

PE Loader (LoadPE)

PE Loader is also often used technique but it's more complicated to code because instead of injecting into a child process, we are injecting in the parent process' memory space. It means that we have resolve IAT, Relocations, TLS Directory, Static TLS (eventually if we aren't calling the entry point by *CreateThread WinAPI*).

PROTECTION

After this research, the author of the paper finds these AVs for the most capable to use:

- Windows Defender (Windows 10)
 - Kaspersky
 - Norton Security
 - ESET Antivirus
 - Avira
- Better than nothing:

Having an Antivirus is not enough but it helps. Human's brain is the strongest protection against malware. Don't open suspicious files unless they are from a trusted source. Scan each downloaded file on <http://virustotal.com> before executing it because they distribute files to AV labs for further analysis and the results appear quickly. Don't look on the detection ratio from infamous AVs because they often mislead to false positive detections. Note that the executables are not the only dangerous files. There are exploits for putting an executable inside a ".doc(x)", ".mp3", ".mp4", ".pdf", etc.

CONCLUSION

This paper presents how easy the AVs can be bypassed. Every injection method could be prevented as like as it was described for Process Hollowing. The author of this paper is thinking of starting an Antivirus Software project that would implement better methods than the existing ones against malware.

REFERENCES

- [1] *Portable Executable*, <https://msdn.microsoft.com/en-us/library/ms809762.aspx>
- [2] *Thread-Local Storage*, https://en.wikipedia.org/wiki/Thread-local_storage
- [3] *Process Environment Block*, https://en.wikipedia.org/wiki/Process_Environment_Block

TREASURELAND – 2D PLATFORM GAME ¹⁴

Iskren Ivanov – Student

Department of Informatics and Information Technologies,
University of Ruse “Angel Kanchev”
Tel.: +359 886165768
E-mail: iskren_damqnov@abv.bg

Assoc. Prof. Katalina Grigorova, PhD

Department of Informatics and Information Technologies,
University of Ruse “Angel Kanchev”
Phone: +359 899458152
E-mail: k.grigorova@gmail.com

Abstract: *Treasureland is a single player 2D platform game. It is based on Unity Engine using C#. Everyone likes playing games in the free time. The games are a good way to relieve stress or just to play them for fun. Treasureland is generally made for children but adults can also play it as well.*

Keywords: *Game, Unity, C#, 2D Platform Game, Game Engine*

ВЪВЕДЕНИЕ

Съществуват много ползи от компютърните игри. Те развиват умения на играещия като: концентрация, рефлексии, логическо мислене, адаптивни умения и др. Treasureland е малка игра и не изисква много производителност на компютъра. Тя е сравнително проста игра и се играе лесно. На фиг. 1 е представено менюто на играта. Основната цел на създателя на Treasureland е създаване на 2D платформена игра на Unity, която включва контролиране на движенията на главния герой в играта и взаимодействие с различни обекти. Този вид игри е любим на много хора. Има много такива игри и въпреки това хората продължават да искат още. Този тип игри е един от най-старите, но продължава да представлява интерес за играчите. Някои ги играят от носталгия, други следят за нови подобрения и нови идеи. Treasureland е създадена с учебни цели.

ИЗЛОЖЕНИЕ

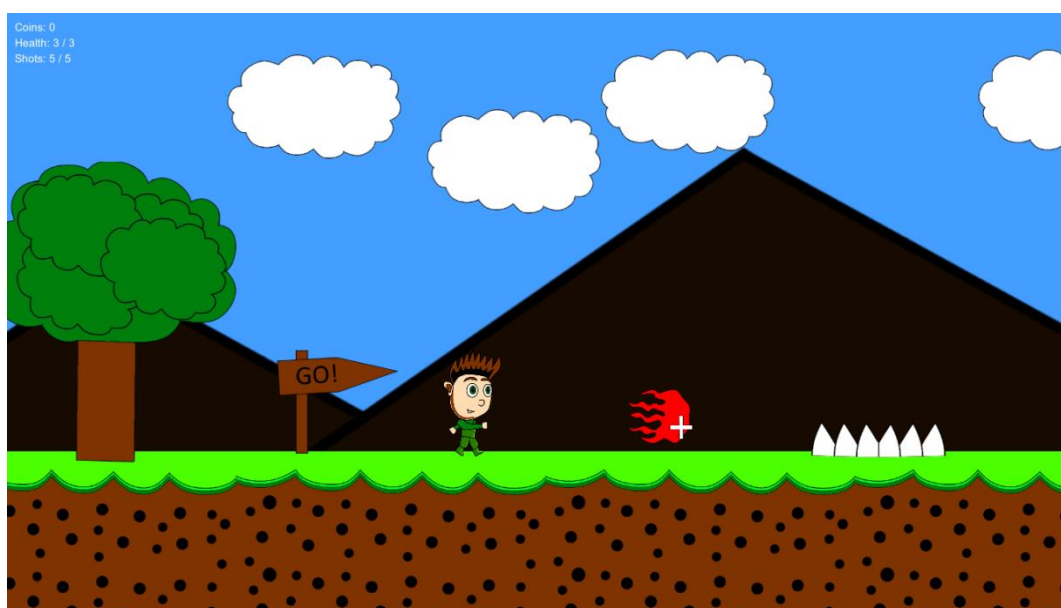
Как работи играта?

Treasureland е максимално опростена. В играта има един главен герой, който играчът контролира. Основните цели в играта са: събиране на съкровища, преодоляване на препятствия и достигане на финала на всяко ниво. Има само 3 нива, но Unity дава възможност изключително лесно да се добавят още, единствено чрез придърпване на обекти във сцената. Главният герой се движи, скача и изстрелва обекти. Играта може да се играе както с клавиатура, така и с джойстик.

¹⁴ Докладът е представен на студентската научна сесия на 03.05.2018 в секция „Информатика“ с оригинално заглавие на български език: TREASURELAND – 2D ПЛАТФОРМЕНА ИГРА.



Фиг. 16. Главно меню



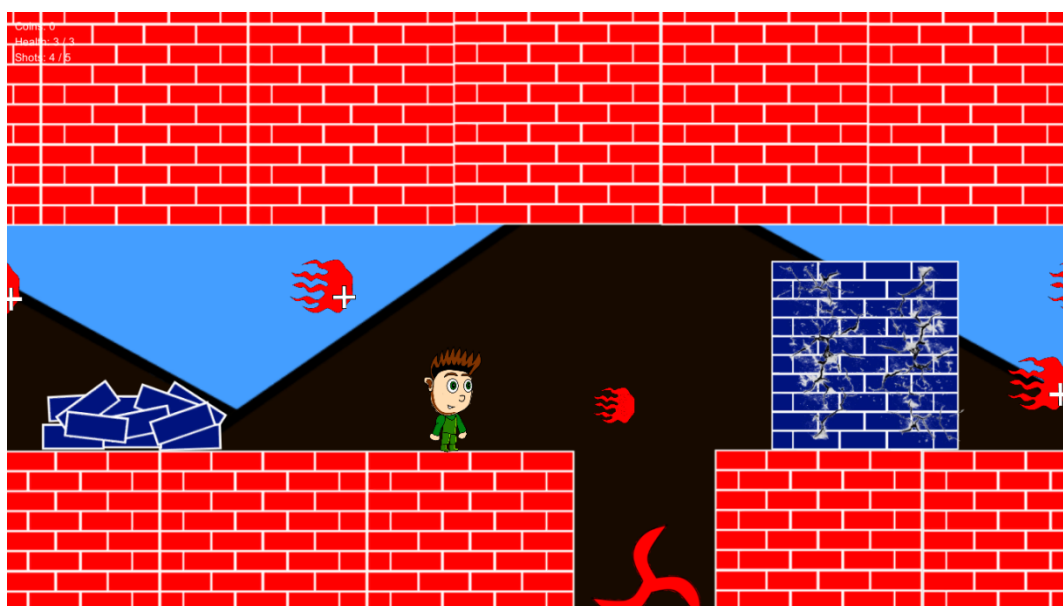
Фиг. 17. Ниво 1

Стартиране на играта

След като играта е стартирана, в горния ляв ъгъл има три съобщения, показващи броя на събраните съкровища (започващ от нула), точките здраве и броя на оставащите изстрели. В началото героят има 3 точки здраве и разполага с 5 изстрела. Той може да унищожава някои препятствия (например стени), стреляйки по тях. Когато броят на изстрелите стане 0, героят не е способен да стреля, докато не се зареди с още изстрели, които могат да бъдат открити на земята. Героят не може да има повече от 3 точки здраве и 5 изстрела, но съществуват обекти, които могат да увеличат максималния брой.



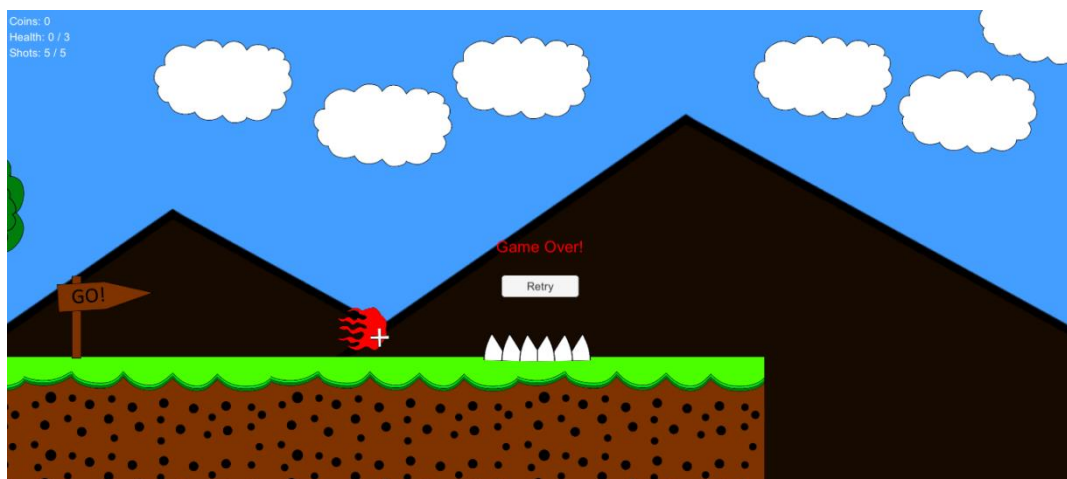
Фиг. 3. Ниво 2



Фиг. 4. Ниво 3

Нива

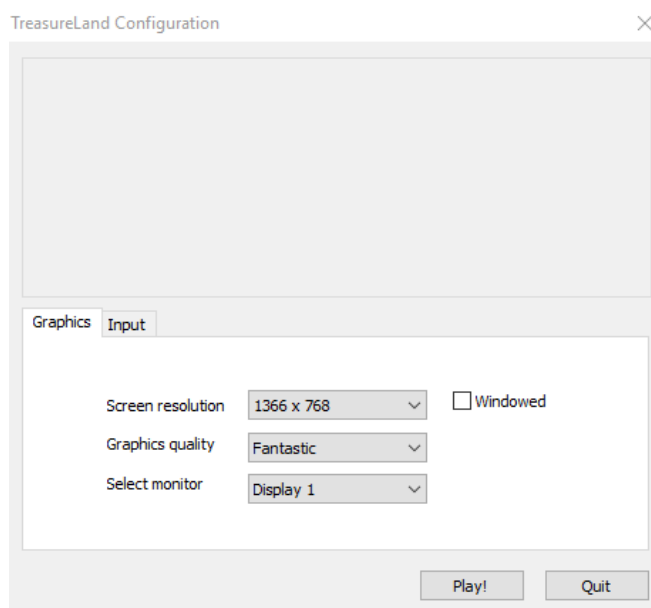
Има три изображения от играта на фиг. 2, фиг. 3 и фиг. 4. Ниво 1 е въвеждащо ниво. На ниво 2 обстановката е малко по-различна. Нощ е и е по-трудно да се видят съкровищата и препятствията. Това прави играта по-предизвикателна и по-интересна. На ниво 3, героят се намира на ново място и той трябва да си проправи път през множество стени.



Фиг. 5. Game Over

Край на играта

Когато героят се докосне до опасност (препятствие), той губи 1 точка здраве. Когато точките здраве станат нула, героят умира и се появява съобщение за рестартиране на нивото. Пропадането в пропаст също води до рестартиране на нивото.



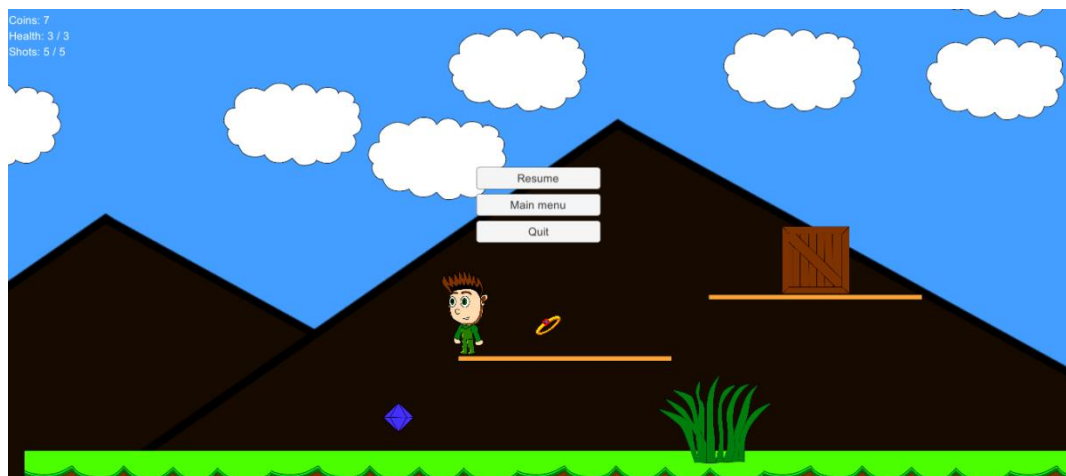
Фиг. 6. Прозорец с конфигурации

Главно меню и меню „пауза“

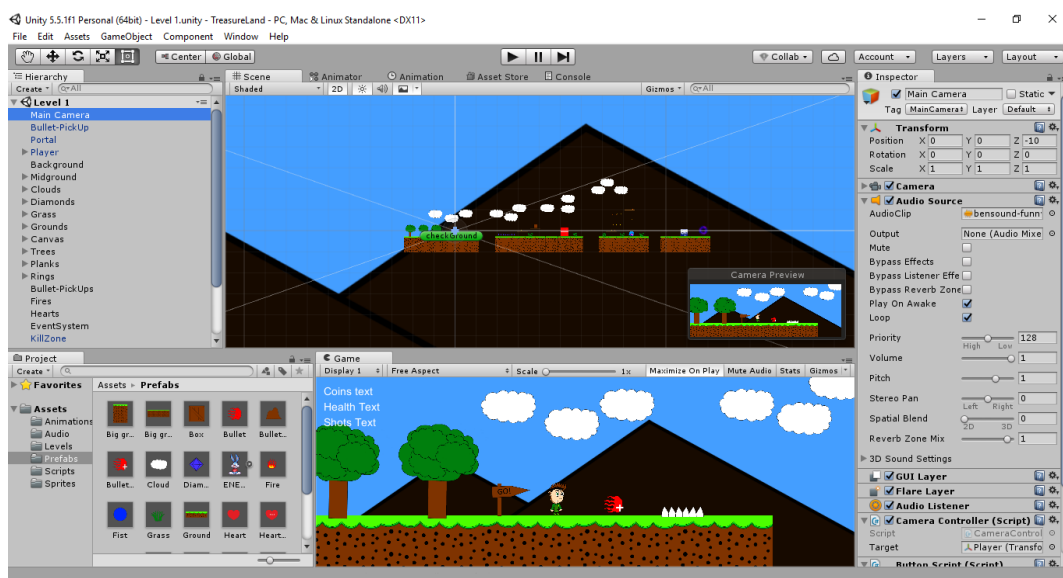
В началото на играта се отваря главното меню, където се избира дали да се стартира някое от нивата или да се излезе от играта. С избиране на ниво – играта започва. С натискане на бутона „Esc“ се отваря меню „пауза“ (фиг. 7). В този момент играта е временно стопирана. Това меню предлага възможност за възстановяване на играта, отиване в главното меню или излизане от играта.

Конфигурации преди стартиране на играта

Unity предлага прозорец, който се отваря преди стартиране на играта. В този прозорец могат да се конфигурират: качеството на графиката, резолюцията, контролите и др.



Фиг. 7. Меню „Пауза“

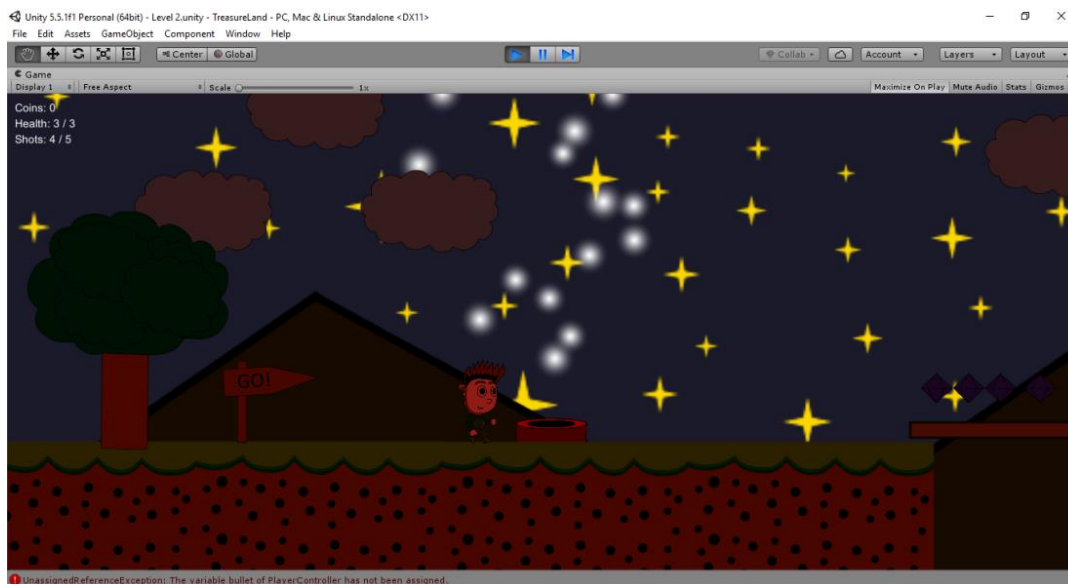


Фиг. 8. Unity платформа

Unity и C#

Правенето на проста игра на Unity [1] е сравнително лесно, но идеите отнемат доста време. Разработката изисква рисуване на изображения, използвайки графичен редактор като Paint.NET, Photoshop и т.н., придърпване на тези изображения в сцената, конфигуриране на техните характеристики и писането на скриптове за тези обекти на C# или Javascript. Когато се добави изображение към сцената, в началото то няма никакви функции. То е просто част от сцената, не му действа гравитация и по никакъв начин не може да се взаимодейства с него. За това е необходимо да се добавят компоненти към изображението (Collider, Rigidbody и др.). След това е възможно да се запази конфигурираното изображение (обект), така че да може да се използва множество пъти. Това прави добавянето на нови нива много лесно. За да се добави функционалност, която не е предоставена от Unity, е необходимо да се напише скрипт. След

това този скрипт се прикача към обекта за който ще действа. Един обект може да има повече от един скрипт. Изображения от енджинът на Unity са представени на фиг. 8 и фиг. 9. На фиг. 8 е изобразен енджинът Unity. Тук се създава играта. Отляво са всички обекти, които са добавени в сцената. Отдясно са всички конфигурации за избрания обект (в този случай – Main Camera). Централно стои самата сцена. Най-отдолу се вижда как изглежда самата игра. На фиг. 9 е изобразен режимът на игра на цял екран. Бутонът „Play” е натиснат и играта е пусната. С натискане на тези бутони играта може да бъде стартирана, спиране или просто паузирана с цел да се конфигурира нещо.



Фиг. 9 Режим „Игра“ в Unity

ИЗВОДИ

Компютърните игри винаги ще представяват интерес за хората. Те непрекъснато се променят и усъвършенстват. Правенето на компютърни игри е добър начин за усвояване на техники от програмирането и компютърната графика. Освен полезно, правенето на игри е и забавно. Това мотивира разработчика да работи редовно и да изпробва нови техники.

REFERENCES

- [1] Unity, <https://unity3d.com/>
- [2] Helpful programming forum, <https://stackoverflow.com>
- [3] Unity tutorials, <http://catlikecoding.com/unity/tutorials/>

TECHNIQUES FOR ANALYSIS OF THE RELIABILITY OF THE SOFTWARE SYSTEMS AND PROBLEMS FOR ENSURING THE RELIABILITY¹⁵

Kamelia Dobрева, PhD Student

Department of Informatics and Information Tehnologies,
“Angel Kanchev” University of Ruse
Tel.: 082-888 470
E-mail: kdobрева@uni-ruse.bg

Assoc. Prof. Plamenka Hristova, PhD

Department of Informatics and Information Tehnologies,
“Angel Kanchev” University of Ruse
Phone: 082-888 326
E-mail: phristova@ami.uni-ruse.bg

Abstract: The paper reviews current trends in research on ensuring the reliability of software systems. The techniques for ensuring the reliability of software systems are analyzed and the problems to use them. Particular attention is paid to measuring the reliability of the software. The objective is to develop an assessment of the reliability of the software in the process of developing requirements or at the design stage of the system. At a later stage, it often leads to a breach of its safety. The main trends in software quality assurance are related to the study of effective software reliability techniques.

Keywords: Software systems, Software quality, Software reliability, Techniques, Problems, Design, Testing, Assessment, Measurement.

ВЪВЕДЕНИЕ

Софтуерът е навсякъде в нашето ежедневие. Той се е превърнал в неразделна част от съвременното модерно общество – домакински уреди, телекомуникации, автомобили, самолети, обучение, развлечения и др. В следствие на това науката и технологиите се нуждаят от висококачествен софтуер. Размерът и сложността на софтуерните системи нараства драстично през последните десетилетия и това е устойчива тенденция. Програмните грешки и ненадеждната работа на системите могат да доведат до сериозни, дори фатални последствия и загуби на бизнеса. Това характеризира надеждността като най-важния измерим фактор за повишаване на качеството и сигурността на софтуерните системи. Софтуерна криза създава огромни възможности за изследователите на софтуерни технологии, както и за практикуващите специалисти. Възможността за производство на качествено софтуерно решение е не само необходимост, но и ключов отличителен фактор за модерния бизнес.

Понятието „софтуерна надеждност“ се дефинира като „вероятността на правилната работа на софтуерната система за определено време в определена среда“ [ANSI/IEEE, 1991]. Сред другите атрибути за качеството на софтуера като функционалност, използваемост, обхватност, поддръжка и т.н., надеждността е основният фактор за качеството на софтуера, тъй като отговаря за способността на софтуера да се справя с грешките по време на изпълнение, и/или способността на алгоритъма да продължи да оперира, независимо от настъпили аномалии във входните данни, изчисленията и т.н.

Съществуващите техники за анализ на надеждността имат своите недостатъци. На първо място, текущите техники събират данните за отказ по време на тестовите за интеграция или

¹⁵ Докладът е представен на студентската научна сесия на 03. 05. 2018 г. в секция „Информатика“ с оригинално заглавие на български език: ТЕХНИКИ ЗА АНАЛИЗ НА НАДЕЖНОСТТА НА СОФТУЕРНИ СИСТЕМИ И ПРОБЛЕМИ ПРИ ОСИГУРЯВАНЕТО НА НАДЕЖНОСТ.

фазите на тестване на системата. Данните за грешки, събрани по време на фазата на късното изпитване, може да са окаже твърде късно за основни промени в проекта. На второ място, данните за неуспех, събирани по време на разработване на системата могат да бъдат взети предвид от разработчиците и по този начин да се избегнат сризове в реална работна среда. Това е особено вярно за висококачествени софтуерни системи, които изискват задълбочено тестване. Трето, настоящите техники или методите за моделиране се основават на някои нереалистични предположения, които правят оценката на надеждността твърде оптимистична спрямо реалните ситуации. Изследването на техники за надеждност на софтуера е необходимо, особено за съвременните софтуерни системи [B. Littlewood and L. Strigini, 2000].

В изложението ще бъдат разгледани основните техники, осигуряващи надеждност на софтуерните системи и ще бъдат описани най-често появяващите се проблеми при използването им. Нивото на отговорност и огромната сложност на софтуерните компоненти, в рамките на различни системи, влияят в голяма степен на тяхната безопасност и надеждно функциониране, което означава, че е изключително важно да се изследва въздействието на софтуера по отношение на надеждността на системите.

ТЕХНИКИ ЗА АНАЛИЗ НА НАДЕЖДНОСТТА

Проучването на техниките за анализ на надеждността на софтуерните системи, води до идентификация и предложение за решение на текущите проблеми при оценяването на надеждността. В бъдеще се очаква да се предложи метод за оценяване на надеждността. Създаването на примерен модел за оценяване на надеждността на софтуерните системи е следствие на разработването на такъв метод за изчисляване, осигуряващ софтуерното качество.

Техниките за анализ на надеждността на софтуерните системи могат бъдат групирани в 3 основни категории:

- *дизайн за надеждност на софтуера* – в първата категория надеждността на софтуерната система се постига чрез разработване на надеждни компоненти на системата. Основен момент в тази категория е да се осигури избягване на грешки и толерантност към грешки. Наличните техники включват откриването, предотвратяването на грешки, тяхното отстраняване и допустимост от появяването им.

- *тестване за надеждност на софтуера* – в тази категорията надеждността на софтуерната система се влияе от техниките за тестване, например тестване на натоварването или регресионно тестване. Основната цел на техниките е да се осигури премахването на неизправности. Техниките включват тестване на потока от данни, тестване на невалидни или неочаквани входни данни и свързаните с тях инструменти.

- *оценка за надеждност на софтуера* – в тази категория надеждността на софтуера се демонстрира най-често посредством използване на техники за моделиране. Основната цел е да се предскаже прогнозирането на грешки. Наличните техники включват задачи за измерване на надеждността на софтуер и инструменти за надеждност на софтуера.

Дизайн на надеждността на софтуера

Разработването на високо надежден софтуер от гледна точка на клиента е трудна работа за всички софтуерни инженери. Michael R. Lyu в своята книга обобщава следните четири технически области, които са приложими за постигането на надеждни софтуерни системи и те могат да се разглеждат и като четири техники [M.R. Lyu (ed.), 1996]:

- *Предотвратяване на грешки*: да се избегнат грешки по време на създаването на софтуера;
- *Отстраняване на грешки*: да се установи чрез проверка и валидиране, наличието на грешки и те да се премахнат;
- *Отклонение при грешка*: да се осигури услуга, отговаряща на спецификацията, независимо от наличието или възникването на грешки;

➤ *Прогнозиране на грешки*: да се оцени наличието на грешки както и събитията и последствията, свързани с тях. Това е основният фокус на моделирането на софтуерната надеждност.

Предотвратяването на грешки е първоначалният защитен механизъм срещу ненадеждност на дадена система. То е цел на всяка методология на софтуерното инженерство. Общите подходи включват формални методи в спецификациите на изискванията и проверки на програмите; ранно взаимодействие с потребителя и усъвършенстване на изискванията; инструменти за проектиране на софтуер; принципи, среди и системни техники за многократно използване на компонентите на софтуера. При успешното им прилагане механизмът може напълно да предотврати грешките. За съжаление обхватът на приложението му е ограничен. Механизмите за предотвратяване на грешки не могат да гарантират избягването на всички софтуерни дефекти. При откриването на грешки в софтуера премахването им е следващото средство за защита. Два практически подхода за отстраняване на грешки са софтуерното тестване и софтуерната инспекция, които са станали стандартни практики в областта на осигуряване на качеството. Въпреки това, понякога остават незабелязани грешки в процесите на тестване и проверка. Те остават във софтуерната система, докато бъдат открити от клиентите. Допустимостта към грешки е последната защитна линия при предотвратяване на появите им. Тя е атрибутът за оцеляване на софтуерните системи по отношение на способността им да предоставят непрекъснато обслужване на клиентите. Техниките за допустимост на програмните грешки позволяват на софтуерните системи:

- (1) да предотвратят появата на пасивни софтуерни бъгове, като защитно програмиране за проверка на условията за въвеждане, извеждане и забрана на неправилни операции;
- (2) да следят появилите се грешки в софтуера с цел да не се разпространяват по-нататък, като рутинни процедури за проверка на неуспешни операции;
- (3) да възстановяват софтуерни операции при погрешни условия, като механизми за проверка и обратно връщане;
- (4) да понасят грешки на ниво система планомерно, като например използването на различни начини на проектиране в разработката на софтуер.

И накрая, ако се очаква да възникнат грешки в софтуера, от съществено значение е те да бъдат оценени и предсказани. Прогнозирането на грешки включва формулирането на връзката виновен-неизправност, разбирането на операционната среда, създаването на модели за надеждност на софтуера, разработването на процедури и механизми за измерване на надеждността на софтуера, както и анализ и оценка на резултатите от измерванията. Възможността да се определи надеждността на софтуера не само ни дава насоки за качеството на софтуера и кога да спрем тестването, но и предоставя информация за нуждите на софтуерната поддръжка. Тя може да улесни валидността на гаранцията на софтуера, когато надеждността на софтуера е била правилно сертифицирана.

Тестване на надеждността на софтуера

Тестването на софтуер е изследване, което се прави с цел да се даде информация на всички заинтересовани страни за качеството на системата. То трябва да даде обективна и безпристрастна оценка на софтуера, която да позволи на бизнеса да прецени и разбере рисковете от употребата му. Целта за тестването на софтуер е да подобри надеждността на софтуера чрез премахване на грешки преди реалната работа със системата. Като типичен механизъм за отстраняване на грешки, тестването на софтуер е широко разпространено в индустрията за осигуряване на качеството и подобряване на надеждността му. Ефективното тестване се дефинира като разкриване на повечето, ако не всички, откриваеми грешки. Техниките за тестване могат да включват изпълнение на програмата с цел откриването на грешки в нея. Тестването може да служи и като метод за статистическа извадка, за да се съберат данни за грешките и да се направи оценка на надеждността на софтуерната система.

Оценка на надеждността на софтуера

Моделирането на надеждността на софтуера привлича все повече вниманието на изследователите както за оценката - измерване на текущото му състояние, така и за прогнозиране - оценка на бъдещото състояние на надеждността на софтуерната система. Моделът за надеждност на софтуера дефинира формата на случаен процес, който описва поведението на програмните повреди по отношение на времето. Исторически преглед, както и перспектива на приложението на моделите за надеждност на софтуера могат да бъдат намерени в [J. Cheng, D.A. Bell, and W. Liu, 1997] и [M.R. Lyu (ed.), 1996]. Съществуват три основни подхода за моделиране на надеждността: подход за зареждане и маркиране на грешки, подход на домейна на данните и подход на домейна на времето, който се счита за най-популярният. Основният принцип на моделирането на надеждността на софтуера в домейна на времето е да се създаде диаграма на наблюдаваните данни във време на отказ по предварително определена формула, така че моделът да може да бъде параметризиран със статистически техники. Такива техники предлагат методите Least Square или Maximum Probability. След това моделът може да предостави оценка на съществуващата надеждност или прогнозиране на бъдещата надеждност чрез по-прецизирани техники. Моделите за надеждност на софтуера обикновено правят редица общи предположения. Такива, например, са:

- (1) Експлоатационната среда, в която трябва да се измерва надеждността, е същата като средата за изпитване, в която е параметризиран моделът за надеждност;
- (2) След като възникне срыв се уточняват причините и той веднага се отстранява;
- (3) Процесът на отстраняване на грешките няма да доведе до нови сривове;
- (4) Броят на грешките, присъщи на софтуера, и начина, по който те се проявяват, така че причиняват сривове, са представени посредством определени математически формули. Тъй като броят на грешките (както и степента на неуспех) на софтуерната система намаляват при тестване, което води до повишаване на надеждността, тези модели често се наричат Модели за повишаване на надеждността на софтуера (Software Reliability Growth Model, SRGM) [M. Jain, T. Manjula, T. R. Gulati, 2011].

Измерване на надеждността на софтуера

Измерването на надеждността на софтуера е процес, включващ анализ на изследванията осигуряващи надеждност на системите и оценяването на грешките, взети от тестване на софтуер. На фигура 1 са показани съществените компоненти на надеждността на софтуера с цел нейното измерване.

- Дефиниране на целите на надеждността на прототипа на софтуерната система - качеството се дефинира количествено от гледна точка на клиента чрез определяне на грешките и тежестта им, чрез определяне на целта за надеждност и чрез уточняване на баланса между ключовите цели за качество (например надеждност, дата на доставка, цена).
- Задаване на ролята на клиент в процеса на изследване на надеждност – създава се така нареченият Оперативен профил, чрез които се определят различните роли и функции в разработката на софтуерната система. Оперативният профил е набор от отделни алтернативи на работата на системата и свързаните с нея вероятности за възникване [M.R. Lyu (ed.), 1996]. Изграждането на оперативен профил насърчава тестери да избират тестови случаи според потребителската употреба на системата, което допринася за по-точна оценка на надеждността на софтуера в тази област.
- Разработване на план на процеса на изследване на надеждност чрез тестове - в тази процедура целите за качество и оперативният профил се използват за управление на ресурсите и насочване на проектирането, внедряването и тестването на софтуер.
- Изпълнение на тестовете - надеждността по време на тестовете се проследява, за да се анализира с цел внедряване и поддръжка на системата.
- Анализ на резултатите от тестовете - надеждността може да бъде анализирана в тази област, за да се валидират усилията за инженеринг на надеждността и да се осигурят обратна връзка за подобряване на етапите на разработка на софтуерната система. В зависимост

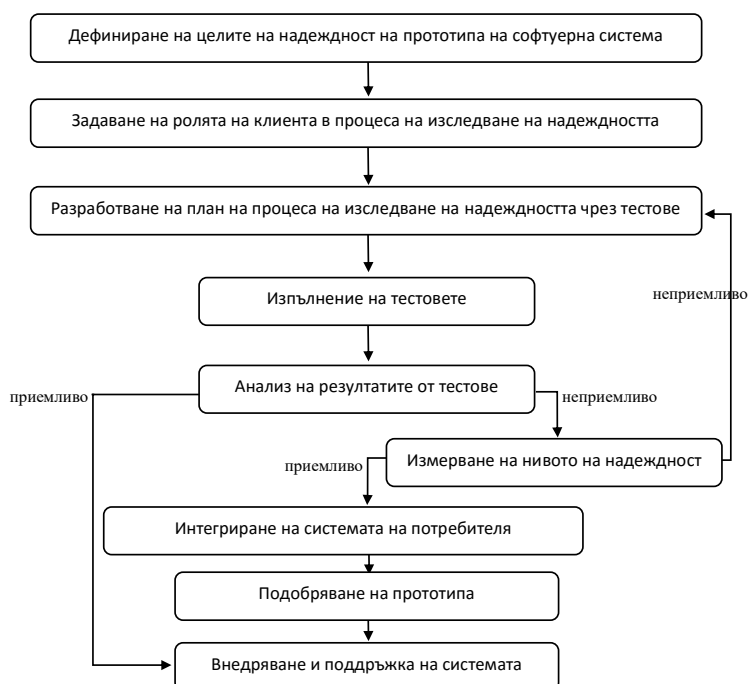
от резултатите се отварят две възможности, първият вариант е с приемливи резултати, при които системата може да се внедри, при втория - резултатите от тестовете, следователно анализа на надеждността са неприемливи. Във втория случай следва измерване на нивото на надеждност.

➤ Измерване на нивото на надеждност – използват се подходящи модели и инструменти за измерване на надеждността на софтуера. Тази дейност може да се повтори, докато не бъде постигнато определено ниво на надеждност, което е приемливо и позволява успешната разработка на прототип

➤ Интегриране на система на потребителя - позволява събиране на ранна информация за експлоатацията, ползваемостта и надеждността на новата система.

➤ Подобряване на прототипа – тази дейност има за цел да повиши нивото на надеждност.

➤ Внедряване и поддръжка – се справят с новооткрити грешки или бъгове след инсталацията на софтуера.



Фигура 1. Процесът на изследване на надеждност

ПРОБЛЕМИ

Предизвикателствата в надеждността на софтуера не само произтичат от големината, сложността, трудността и новостта на софтуерните приложения в различни области, но също така се отнасят до знанията, обучението и опита на участващите разработчици. Проблемите на надеждността на софтуера, разгледани на ниво системна реализация и проектиране, са описани в таблица 1.

За да се измери надеждността на софтуера, тя трябва да се прогнозира още на етапите проектиране и програмиране на разработваната софтуерна система. Подобряването на надеждността на софтуера е трудна задача. Трудността на проблема произтича от това, че оценката за надеждност и безопасност се изготвя след реализацията на дадена система. Констатирането на грешки в дизайна или изпълнението водят до нарушаване на безопасността и надеждността, както и до скъпи и ресурсоемки процеси. Преодоляването на този етап е свързано с изследване въздействието на софтуера по отношение на безопасността на системите.

Таблица 1. Проблеми, свързани с надеждността на софтуера

Проблем	Решение
Често подобни системи и дори различни версии на една и съща може да са базирани на различни спецификации, затова различните софтуерни системи се явяват „една от вид“.	Да се направи оценка на тежестта на възможните проблеми и избор на подходящ статистически метод във всяка спецификация да се вземе решение.
Зависимост от хардуерните компоненти, които влияят на софтуерните такива.	При изграждането на система с висока надеждност да се предвидят решения, които да дефинират подходящи подмени на отказите на хардуерните компоненти.
Тестерите следват интуитивен подход при проверка на работоспособността на софтуера.	Въвеждане на задължителното ползване на подходящи метрики за покритие на кода и други части на проекта.
Недостатъчно изчерпателно формулирани функционални и нефункционални изисквания към системата	Въвеждане на подходящи критерии, които в процеса на разработка на изискванията, гарантират високо ниво на надеждност
Изискване за безотказна работа без толеранс на грешки, ограничава възможностите за статистически анализ на надеждността	
Противоречиви изисквания към модули, които са взаимосвързани	

ЗАКЛЮЧЕНИЕ

Надеждността е една от най-важните характеристики, присъщи на концепцията “качество на софтуера”. Тъй като разходите за сринове на софтуерни приложения нарастват и тъй като грешките все повече оказват влияние върху производителността на бизнеса, надеждността на софтуера придобива особено важно място в процеса на разработване и оценка на качеството на софтуерните системи. Грешките в софтуера се дължат на пропуски в процеса на проектирането и разработването му, затова основната задача е усъвършенстване на тези процеси. Решението на това софтуерните системи да функционират правилно и да отговарят на изискванията на клиентите е да се извърши измерването на надеждността по време на ранните етапи на разработката им. Използването на ефективни техники за анализ и оценка на надеждността на софтуерните системи, които допринасят за подобряване на процесите, осигуряващи качеството на софтуерните системи, е цел на софтуерната индустрията, както и голямо предизвикателство на софтуерните разработчици.

REFERENCES

- ANSI/IEEE, *Standard Glossary of Software Engineering Terminology*, STD-729-1991, ANSI/IEEE, 1991.
- J. Cheng, D.A. Bell, and W. Liu, “*Learning Belief Networks from Data: An Information Theory Based Approach*”, Proceedings of the Sixth International Conference on Information and Knowledge Management, Las Vegas, 1997, pp. 325-331.
- B. Littlewood and L. Strigini, “*Software Reliability and Dependability: A Roadmap*”, in Proceedings of the 22nd International Conference on Software Engineering (ICSE'2000), Limerick, June 2000, pp. 177-188.
- M.R. Lyu (ed.), *Handbook of Software Reliability Engineering*, IEEE Computer Society Press and McGraw-Hill, 1996.
- M. Jain, T. Manjula, T. R. Gulati, “*Software Reliability Growth Model (SRGM) with Imperfect Debugging, Fault Reduction Factor and Multiple Change-Point*” in Proceedings of the International Conference on Soft Computing for Problem Solving (SocProS 2011), Roorkee, India, December 20-22, 2011, pp 1027-1037

USING THE METHODS OF CLUSTERING AND CLASSIFICATION FOR PROJECT TASK EXECUTION TIME PREDICTION¹⁶

Natalia Bobrovskaya – Student

Department of Innovation and High Technology,
Moscow Institute of Physics and Technology, Russia
E-mail: nataliya.bobrovskaya@phystech.edu

Pavel Akhmetchanov

Chief Programmer of the Web Interface Department,
Tinkoff Bank, Russia
E-mail: pavel.power@gmail.com

Abstract: In the article questions of performance time prediction and story points estimation on the basis of numerical and text attributes are considered. To evaluate the significance of words in a set of documents Tf-idf metric is used. Clusters of tasks in terms of time and story points are defined using the T-SNE algorithm. The logistic regression algorithm for story points classification is used.

Keywords: clustering, classification, estimation, metrics, data set, text attributes, story points

ВВЕДЕНИЕ

При создании программного обеспечения с использованием гибких методологий разработчики постоянно сталкиваются с неполной и изменяющейся ситуацией: меняются наборы задач, задачи могут появляться или исчезать по ходу выполнения проекта, часть параметров уточняется по мере выполнения задачи, могут отсутствовать данные об исполнителях задачи, о ее связях с другими задачами, часть информации является строго конфиденциальной и доступ к ней строго ограничен.

Одним из основных параметров является время выполнения задачи. Как правило, предполагаемое время выполнения задачи определяется исходя из объема задачи. Объем задачи определяется на этапе оценки, которая производится либо всей командой, либо опытными разработчиками.

Целью данного исследования является оценка времени выполнения задач нового проекта на основе данных предыдущих проектов. Данные о задачах взяты из системы отслеживания задач jira IT-подразделения одной из финансовых компаний. Объем выборки составил около 2000 задач. Для решения задачи было разработано программное обеспечение на языках JavaScript (в том числе npm библиотеки jira-client, moment, Az) и Python (библиотеки numpy, pandas, matplotlib, sklearn, scipy). Для обработки данных использовались кластеризация, Tf-idf - метрика, алгоритм машинного обучения t-SNE, алгоритм логистической регрессии.

ПРОГНОЗИРОВАНИЕ ВРЕМЕНИ ВЫПОЛНЕНИЯ ЗАДАЧИ ПРОЕКТА

Жизненный цикл проекта

Под проектом в дальнейшем будем понимать совокупность задач, которые необходимо выполнить для реализации некой функциональности. На Рис.1 изображена типовая схема жизненного цикла проекта.

¹⁶ Докладът е представен на студентската научна сесия на 03. 05. 2018 г. в секция „Информатика“ с оригинално заглавие на руски език: ПРОГНОЗИРОВАНИЕ ВРЕМЕНИ ВЫПОЛНЕНИЯ ЗАДАЧИ ПРОЕКТА МЕТОДАМИ КЛАСТЕРИЗАЦИИ И КЛАССИФИКАЦИИ.

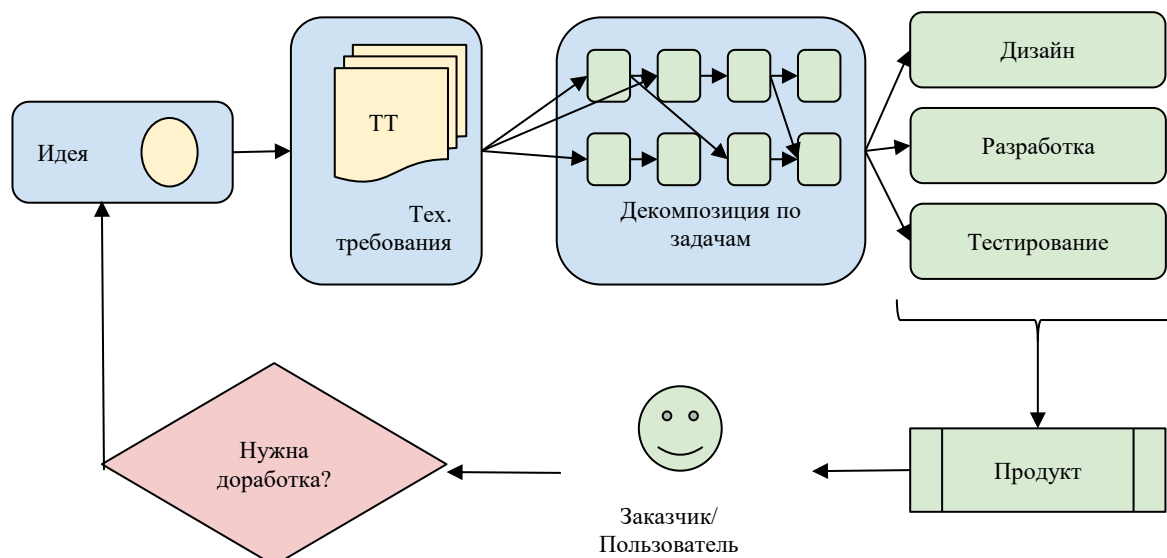


Рис.1 Жизненный цикл проекта

В начале разработки формируются технические требования ТТ, основанные на идее реализации функционала. После этого производится декомпозиция требований на задачи. Временем работы над проектом считается время, начиная с момента начала выполнения первой задачи проекта и заканчивая моментом завершения последней задачи (в том числе её тестирования). Готовый продукт передается для использования Заказчику/Пользователю. Если проекту нужна доработка (новая функциональность) – создается новый проект с новыми техническими требованиями.

Состав данных и подготовка датасета

Данные о задаче включают: название, описание, время начала, время конца, название спринта, название эпика, информацию о связанных задачах, информацию о изменении состояний задачи, тип задачи, название команды разработчиков.

Спринт - это ограниченная по времени итерация проекта, включающая в себя совокупность задач, которые должны быть выполнены за фиксированное время, эпик - это бизнес-постановка задачи (User Story), требующая декомпозиции на технические задачи. Информация о связанных задачах включает в себя имена задач, с которыми связана текущая задача и тип связи. Время выполнения задачи начинается с момента перехода в состояние “In progress” и заканчивается временем перехода в состояние “Done” без учета праздников и выходных дней. Объем задачи может быть оценен в Story Point. Story point (SP) [2,3] - это последовательность чисел, чаще всего чисел Фибоначчи, которая позволяет провести сравнительную оценку. У команд с историей разработки есть документация, которая позволяет оценить задачу в SP по критериям сути задачи, текущим наработкам по ней, сложности интеграции и другим.

Для формирования кластеров был сформирован датасет признаков (метрик) по имеющейся информации о задаче. Данные должны быть преобразованы в датасет числовых признаков. Количество спринтов, количество эпиков, количество связанных задач по каждой связи являются числовыми признаками, тип задачи и название команды обрабатываются как категориальные признаки со значением 0 или 1 в соответствующей колонке в датасете. Для текстовых признаков была проведена обработка текста, построен словарь значимых слов [1] с помощью словаря были сформированы датасеты. В качестве метрики был выбран tfidf признак [5].

Кластеризация

Следующим этапом исследования являлось выделение из существующего набора задач кластеров с похожими характеристиками. Для кластеризации используется t-SNE (t-distributed stochastic neighbor embedding) алгоритм [6]. Для проведения экспериментов использовалась библиотека sklearn языка Python.

Эксперимент 1. Кластеризация с учетом категориальных признаков

Была проведена кластеризация с учетом всех признаков, в том числе категориальных признаков названия команд и типа задачи. Результаты эксперимента представлены на рисунке 2, каждой задаче соответствует одна точка.

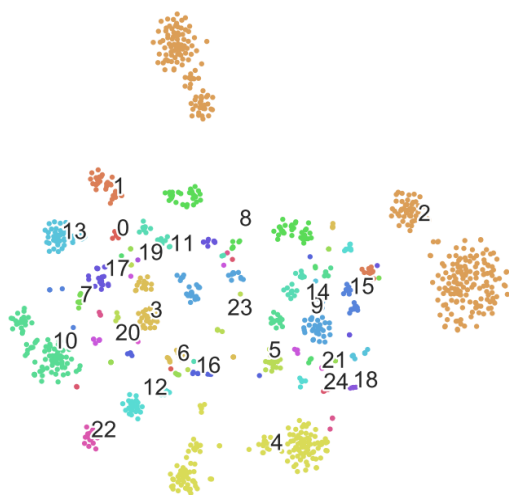


Рис.2(а)

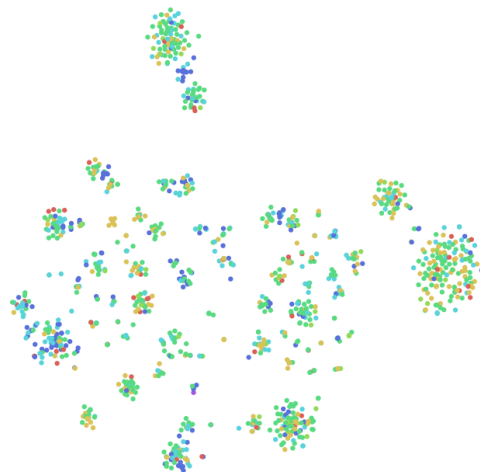


Рис.2(б)

На рисунке 2(а) задачи разных команд изображены разными цветами. Видно, что кластеризация проводится преимущественно по признакам названия команд и типу задачи. Номера команд поставлены в центре выделенных кластеров, дробление внутри команд соответствует различным типам задач. На рисунке 2(б) разными цветами изображены задачи с разным временем выполнения. Красным цветом отмечены те, на которые потрачено меньше всего времени (несколько часов), далее по цветам радуги, фиолетовым - задачи, на которые затрачено много времени (около месяца). При сравнении рис.2(а) и рис.2(б) можно заметить, что задачи команды #10 занимают много времени, а команды #22, #12, #6 имеют преимущественно малые по времени длительности задачи. Следует иметь в виду, что скорость выполнения задач может быть разной из-за различного экспертного опыта каждой команды.

Эксперимент 2. Кластеризация без учета категориальных признаков

В данном случае в выборке были оставлены только количественные признаки и сформированные из текста данные. Результаты эксперимента представлены на рисунке 3.

На рисунке 3(а) разными цветами окрашены задачи разных команд. Некоторые наборы задач команд, например, #2, #22, #23, #16, #0, #17, #15 заметно располагаются в стороне от остальных. Хотя признаки команд не были добавлены в выборку - выборка локализуется по ним. Значит, остальные признаки неявно связывают задачи выполненные одной командой. Например, команда #2 может использовать слова в описании задач, которые не встречаются в задачах других команд.

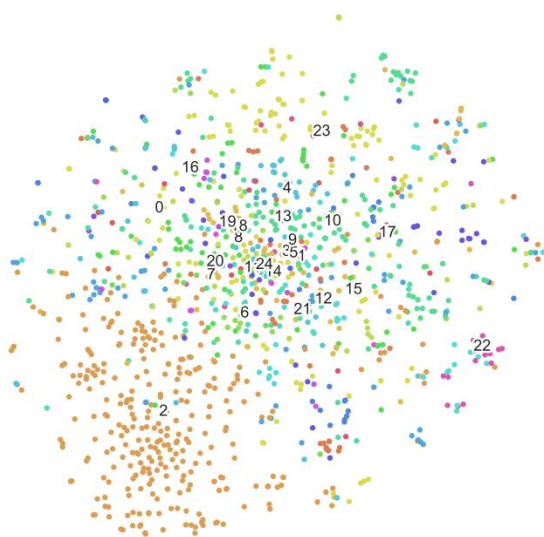


Рис.3(а)

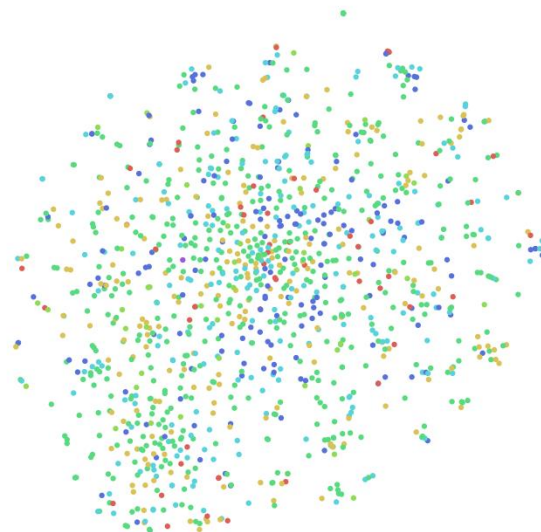


Рис.3(б)

Рисунок 3(б) аналогичен рисунку 3(а) только вместо окраски по командам используется окраска по времени выполнения. Как и на рисунке 2(б), красным цветом отмечены те, на которые потрачено меньше всего времени далее по цветам радуги. В рамках всех команд сложно сделать какие-либо выводы по времени. Возможно, кластеры малы и их достаточно много.

Эксперимент 3. Анализ задач одной команды с учетом Story Point

Метрика Story Point (SP) не была задействована для выборки по всем командам, потому что не все команды ей пользуются. В команде #2 она используется, поэтому выборка по команде #2 была рассмотрена дополнительно. Размер выборки 467 строк. Результаты кластеризации по выборке задач команды #2 представлены на рисунке 4.

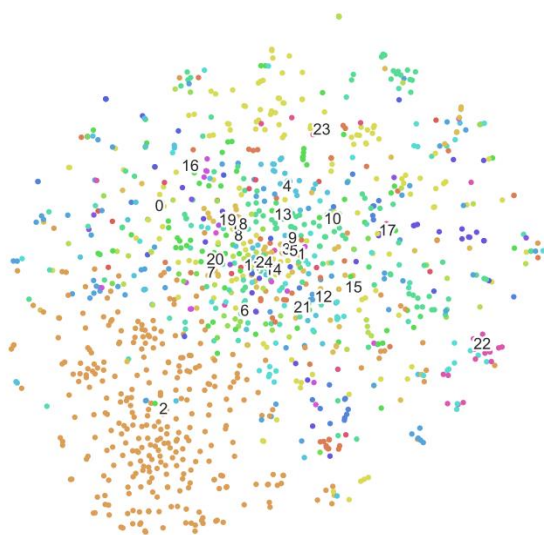


Рис.4(а)



Рис.4(б)

На рисунке 4(а) цветом выделено время выполнения задачи и подписано количество дней, на рисунке 4(б) цветом выделены Story point (SP) и подписано количество SP. При сравнении двух рисунков можно заметить, что:

- близкие SP расположены вблизи друг друга, и большие отделены от малых,
- время связано с SP, но не линейно,
- явного разделения на кластеры нет.

Для изучения связи SP и времени выполнения задачи были построены лепестковые диаграммы (рисунок 5).

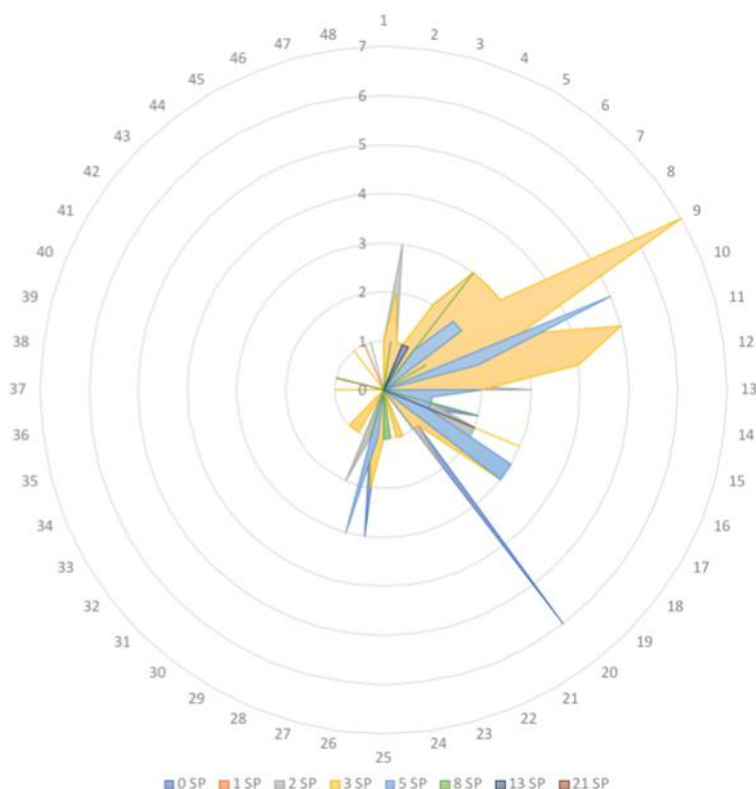


Рис.5 Количество SP относительно времени выполнения задач

Из лепестковой диаграммы видно, что задачи, оцененные в 1 SP, могут выполняться как 9 дней, так и 45. Такое различие в оценке объема задачи (SP) и времени её выполнения может объясняться тем, что:

- выполнение одной задачи зависит от другой, может, даже находящейся в разработке у другой команды,
- SP выставлен неправильно,
- из-за высокого SP были задействованы многие ресурсы для успешного решения задачи, следовательно она завершилась быстрее, чем могла,
- выполнение задачи может зависеть от разных рисков, которые не учли при начале работы над задачей.

Была выдвинута гипотеза, что объем выполнения задач можно предсказать по описанию задачи и другим ее метрикам.

Для подтверждения этой гипотезы была проведена классификация задач по SP для тех команд, которые используют SP и определен алгоритм классификации для оценки объем новых задач, что позволит с учетом ресурсов команды определить время выполнения задачи.

Классификация

Классификацию по SP в рамках команды #2 была проведена с использованием алгоритма логистической регрессии [7]. Для проведения экспериментов использовалась библиотека sklearn языка Python (алгоритм LogisticRegression из sklearn.linear_model).

Эксперимент 4. Классификация с использованием логистической регрессии

Имеющийся датасет несколько раз поделили на обучающую (90%) и тестовую выборку (10%), обучили модель и определили точность предсказаний.

Результаты одной из тестовых выборок представлены на рисунке 6.

real	pred	days	prob:[	0	1	2	3	5	8	13	21]
3	3	3	[0.02,	0.02,	0.17,	0.48,	0.25,	0.04,	0.01,	0.01]	
3	5	25	[0.05,	0.01,	0.05,	0.37,	0.37,	0.11,	0.02,	0.01]	
13	5	45	[0.01,	0.01,	0.02,	0.27,	0.46,	0.19,	0.03,	0.02]	
13	3	9	[0.01,	0.02,	0.02,	0.44,	0.27,	0.17,	0.04,	0.02]	
5	3	7	[0.03,	0.03,	0.1,	0.34,	0.32,	0.13,	0.03,	0.02]	
8	3	1	[0.04,	0.03,	0.24,	0.48,	0.1,	0.08,	0.01,	0.01]	
3	3	0	[0.01,	0.08,	0.28,	0.49,	0.09,	0.03,	0.01,	0.01]	

Рис.6 Результаты классификации одной из тестовых выборок

Первая колонка таблицы - реальное количество выставленных SP, вторая колонка – количество SP, предсказанное алгоритмом, третья колонка - количество дней, затраченных на выполнение задачи, и четвертая колонка - предсказанные алгоритмом логистической регрессии вероятности для каждого SP. Видно, что при выставленных 13 SP задачи выполняются 9 и 45 дней, а алгоритм разграничивает эти две задачи на разные SP: 5 SP для 45 дней и 3 SP для 9 дней. Также видны примеры, когда для реальных 5 SP задача выполняется 7 дней, а для 8 SP - 1 день. Это связано с тем, что SP не имеют абсолютной величины, это сравнительная мера - задачи, схожие между собой по объему, имеют одинаковые SP. Точность предсказаний для разных обучающих и тестовых выборок варьируется от 30 до 60%.

На рисунке 7 показана зависимость между временем выполнения задачи и ее SP.

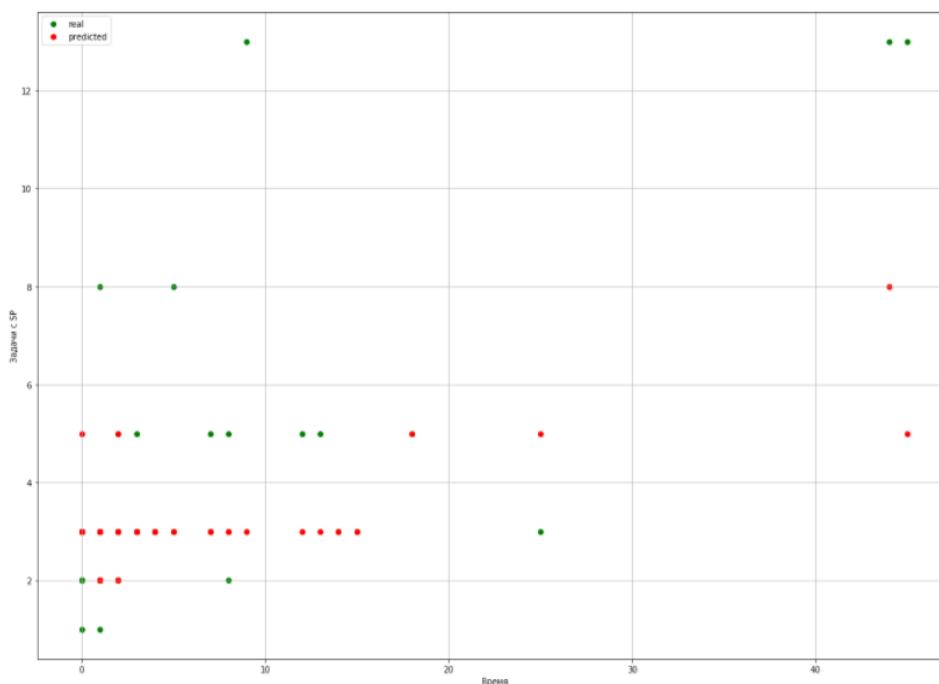


Рис.7 Зависимость между временем выполнения задачи и ее SP

По оси x отложено время выполнения задачи, по оси y SP: зелеными точками отмечены реальные SP для задачи, красными предсказанные алгоритмом. Видим, что красные точки расположены более линейно по времени, а для зеленых точек характерны выбросы, например, задача оцененная в 13 SP выполнялась менее 10 дней.

Исходя из того, что алгоритм предсказывает SP, основываясь на сравнении с существующими задачами, было выдвинуто предположение, что алгоритм может предсказывать лучше, если убрать выбросы из выборки.

Эксперимент 5. Классификации по выборкам без выбросов

По каждому SP был построен график распределения по времени выполнения задачи (в часах), исходя из предположения, что распределение нормальное[4].

Результат для некоторых SP приведен на рисунке 8.

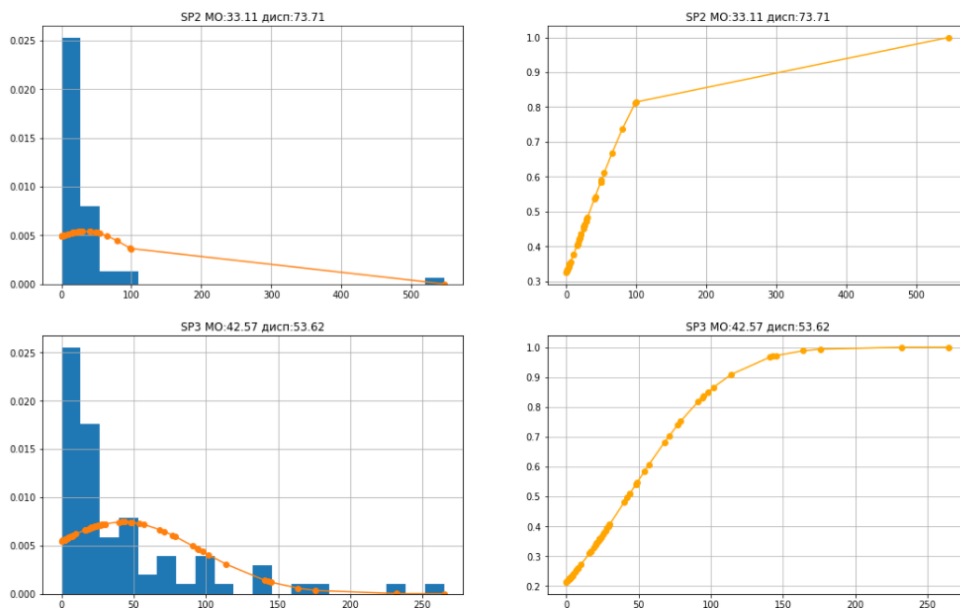


Рис.8 График распределения по времени выполнения задачи

Для каждого SP был задан интервал эталонных задач, и при классификации учитывались только такие задачи. График распределения по времени выполнения задачи для некоторых SP по отфильтрованным выборкам приведен на рисунке 9.

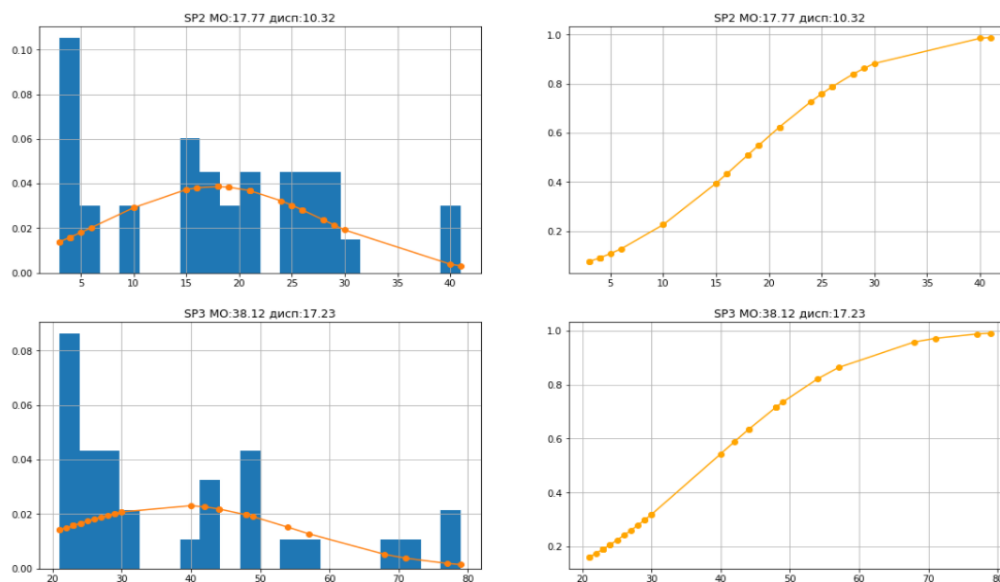


Рис.9 График распределения по времени выполнения задачи отфильтрованным выборкам

По отфильтрованным задачам был построен алгоритм классификации с использованием логистической регрессии. На рисунке 10 приведены результаты по одной из тестовых выборок. В первой колонке приводится SP, предсказанные алгоритмом, во второй - предсказанные алгоритмом логистической регрессии вероятности для каждого SP (1SP, 2SP, 3SP, 5SP, 8SP), третья колонка - реальное количество выставленных SP.

	predicted	prob	real
	3.0	[0.1, 0.33, 0.46, 0.07, 0.04]	5.0
	3.0	[0.03, 0.19, 0.48, 0.19, 0.11]	2.0
	3.0	[0.1, 0.37, 0.41, 0.09, 0.03]	3.0
	3.0	[0.06, 0.38, 0.47, 0.06, 0.03]	1.0
	2.0	[0.06, 0.64, 0.26, 0.02, 0.03]	2.0
	2.0	[0.05, 0.53, 0.17, 0.16, 0.1]	1.0
	2.0	[0.11, 0.49, 0.31, 0.06, 0.03]	2.0
	3.0	[0.29, 0.27, 0.4, 0.03, 0.02]	3.0
	2.0	[0.12, 0.68, 0.14, 0.03, 0.02]	2.0
	2.0	[0.11, 0.38, 0.25, 0.2, 0.06]	2.0

Рис.10 Результаты классификации по измененному алгоритму

Точность классификации (ассигасу) вычисляется как отношение количество правильно классифицированных объектов к количеству всех объектов. По тестовой выборке, представленной на рисунке 10, точность классификации составила 60%, по остальным тестовым выборкам точность классификации варьируется от 40% до 80%. Зная объем SP и исходя из предположения о нормальном распределении времени выполнения задачи можно оценить время выполнения задачи и интервал, в котором оно может находиться.

ВЫВОДЫ

По результатам исследования можно сделать следующие выводы:

- команды различаются между собой опытом и постановкой процесса разработки, поэтому время выполнения задач с одинаковым объемом может различаться;
- кластеризация в рамках всех команд не дает возможность оценить время выполнения задачи;
- классификация с использованием логистической регрессии позволяет оценить объем и время выполнения задачи;
- удаление выбросов из выборки улучшает прогноз объема задачи и времени ее выполнения.

REFERENCES

- [1] Mueller Andreas C., Guido Sarah. *Introduction to Machine Learning with Python*. O'Really Media, 2016
- [2] Kniberg Henrik, Skarin Matthias. *Kanban and Scrum - Making the Most of Both*. Stockholm, published on site <http://www.infoq.com/minibooks/kanban-scrum-minibook> - 2009-12-21- 41p., ISBN: 978-0-557-13832-6.
- [3] Jeff Sutherland and Scrum, Inc. *Scrum. Revolutionary project management techniques*, 2014.
- [4] Cohn Mike. *How Do Story Points Relate to Hours?* 2009 - <http://www.mountangoatsoftware.com/blog/how-do-story-points-relate-to-hours>
- [5] Henrik Brink, Joseph W. Richards, Mark-Fetherolf. *Real-World Machine Learning*. ISBN 10: 1617291927 ISBN 13: 9781617291920.
- [6] van der Maaten, L.J.P.; Hinton, G.E. *Visualizing High-Dimensional Data Using t-SNE*. Journal of Machine Learning Research 9:2579-2605, 2008
- [7] Hastie, T., Tibshirani, R., Friedman, J. *The Elements of Statistical Learning*, 2nd edition. — Springer, 2009. — 533 p.

JEWELRY DESIGN STUDIO DATABASE DEVELOPMENT¹⁷

Yuliya Vlasova – Student

Applied Informatics in Economics Department,
Volga Region State University of Service, Togliatti, Russia
Tel.: +7 8482 22 9108
E-mail: e-mail: jdvlasova@yandex.ru

Assoc. Prof. Elena Malysheva, PhD

Applied Informatics in Economics Department,
Volga Region State University of Service, Togliatti, Russia
Tel.: +7 8482 22 9108
E-mail: em_tgas@mail.ru

Abstract: *In the article features of database development of the specialized information system intended for jewelry design studio are considered. A semantic approach was applied, semantic network and data model of the jewelry manufacturing process were created. Applying the semantic network model to the design studio databases development allowed to take into account the subject area peculiarities and the jewelry design activities specifics*

Keywords: *information system, database development, semantic network, subject area, jewelry design, individual order, jewelry, information support*

ВВЕДЕНИЕ

На сегодняшний день в сфере дизайна наблюдается дефицит информационных систем, которые учитывали бы особенности технологических и информационных процессов, происходящие на стадии дизайна продукта. В дизайн-студиях процессы общения с клиентом и организация работ налажены не так четко, как на производстве. Это связано с особенностью самого понятия дизайна - во многом это творческий процесс, который довольно сложно документировать и стандартизировать. Важным фактором, который нужно учитывать, является удовлетворение клиента привлекательным дизайном в короткий промежуток времени. Актуальной является разработка информационной системы, которая учитывала бы специфику дизайна ювелирных изделий, потому что при современном развитии технологий производства, процесс создания концепции и разработка дизайна модели зачастую занимает больше времени, чем процесс изготовления самого ювелирного изделия. Специализированная информационная система должна иметь базу данных, адекватно отражающую особенности предметной области.

ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

В статье рассматривается проектирование базы данных информационной системы студии ювелирного дизайна, приоритетным направлением деятельности которой является изготовление ювелирных изделий по индивидуальным заказам. В связи с этим особую роль в деятельности студии играет дизайн-бюро – важное структурное подразделение фирмы, для которого и разрабатывается специализированная информационная система. В дизайн-бюро работают сотрудники, занимающие следующие должности: дизайнер, 3D-моделлер, оператор 3D-принтера, маркетолог, менеджер. Также у студии есть ювелирное производство, на котором работают главный ювелир (технолог) и мастера-ювелиры, а также отдел технического контроля, склад и бухгалтерия. Основное управление производит директор, которому подчиняются все отделы.

¹⁷ Докладът е представен на студентската научна сесия на 03.05.2018 в секция „Информатика“ с оригинално заглавие на руски език: ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ СТУДИИ ЮВЕЛИРНОГО ДИЗАЙНА.

Для проектирования базы данных индивидуального заказа на изготовление ювелирного изделия была создана модель семантической сети [2-4,7,9,10] данного процесса, отражающая знания о процессах и изделии (а также самом заказе), которые в ходе выполнения заказа постепенно накапливаются и конкретизируются.

Семантическая сеть изготовления ювелирного изделия по индивидуальному заказу показана на рис.1.

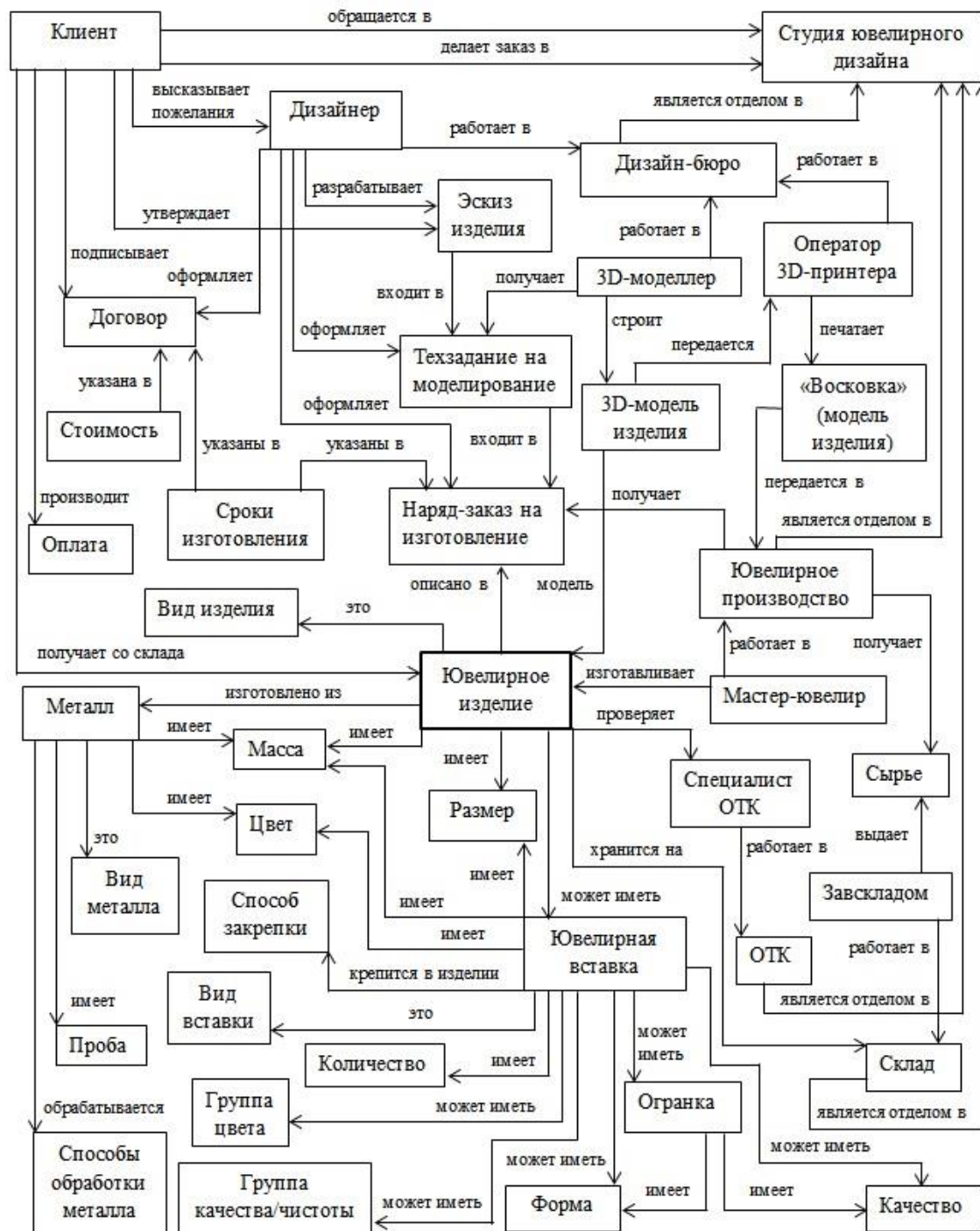


Рис.1. Семантическая сеть процесса «Изготовление ювелирного изделия по индивидуальному заказу в студии ювелирного дизайна»

Процесс выполнения индивидуального заказа начинается с обращения клиента в ювелирную фирму с целью заказать изделие с индивидуальным дизайном. Клиент общается с дизайнером (сотрудником дизайн-бюро), высказывает ему свои пожелания к изделию и на их

основе дизайнер разрабатывает дизайн изделия (ручной эскиз изделия на начальном этапе), который должен утвердить клиент. Далее дизайнер составляет техническое задание на моделирование, 3D-моделлер строит по нему 3D-модель изделия в специальной программе и передает файл для печати на 3D-принтере, распечатанная восковая модель передается на производство. Дизайнер составляет договор с клиентом, в котором указываются стоимость и сроки изготовления изделия, клиент его изучает и подписывает, производит оплату. После этого мастер-ювелир получает от дизайнера наряд-заказ на изготовление изделия (включающий эскиз изделия, проекции 3D-модели и описание характеристик изделия – количество металла, какой металла, какие вставки и сколько, а также сроки изготовления). Производство получает со склада поступает необходимое сырье и изготавливает изделие. Готовое ювелирное изделие проверяет на соответствие специалист отдела технического контроля (ОТК) и после этого изделие поступает на склад, откуда его забирает клиент.

Построенная модель семантической сети достаточно полно отражает основные взаимосвязи и понятия предметной области ювелирного дизайна, что позволит использовать данную модель как основу для разработки базы данных специализированной информационной системы [1,5].

База данных информационной системы [6,8,11,12], спроектированная для студии ювелирного дизайна, имеет свою особенность – большая ее часть занята таблицами, относящимися к описанию самого ювелирного изделия. Основное внимание направлено на отображение всей информации об изделии, как об уникальном продукте. Отличительной особенностью является необходимость хранить большое количество файлов различного формата, особенно изображений. К хранимым данным можно отнести такие категории как, аналоги будущих изделий, предварительные эскизы, утвержденные эскизы изделий, заполненные формы техзаданий и наряд-заказов, файлы трехмерных моделей и проекции этих моделей, и наконец, фотографии готовых изделий. Все это хранится в таблице, описывающей внешний вид изделия. Схема базы данных информационной системы представлена на рис.2.

Для схемы базы данных студии ювелирного дизайна характерна древовидная структура: к таблице с изделиями идут уточняющие таблицы с описанием металла в изделии, вставок, застежек, внешнего вида изделия и т.д. В свою очередь таблица металла в изделии имеет свои зависимые таблицы, описывающие вид металла, его цвет и пробу, а также виды обработки. Таблица, описывающая вставки в изделии связана с таблицами – вид и название вставки, цвет, качество, вид закрепки вставки, а связанная со вставкой таблица огранки в свою очередь связана с таблицами формы и названия огранки, качества огранки и т.д. Сложное изделие может иметь очень обширное описание, в его производстве может быть использовано несколько видов металлов, множество различных вставок, каждая со своими характеристиками.

Описание уникального изделия – это комбинация не только множества индивидуальных эстетических характеристик, но и примененных конструкторских решений, описаний компонентов, обладающими конкретными физическими свойствами. Все это дает основания полагать, что для сферы дизайна самыми востребованными могут стать именно специализированные информационные системы с обширными базами данных, спроектированными с акцентом на описание изделия. Помимо знаний о заказах и изготовленных изделиях, база данных также должна содержать обширную информацию о предметной области – классификации различного сырья, используемого при производстве ювелирных изделий, правила оценки ювелирных изделий и вставок, описания конструкторских решений и т.д.

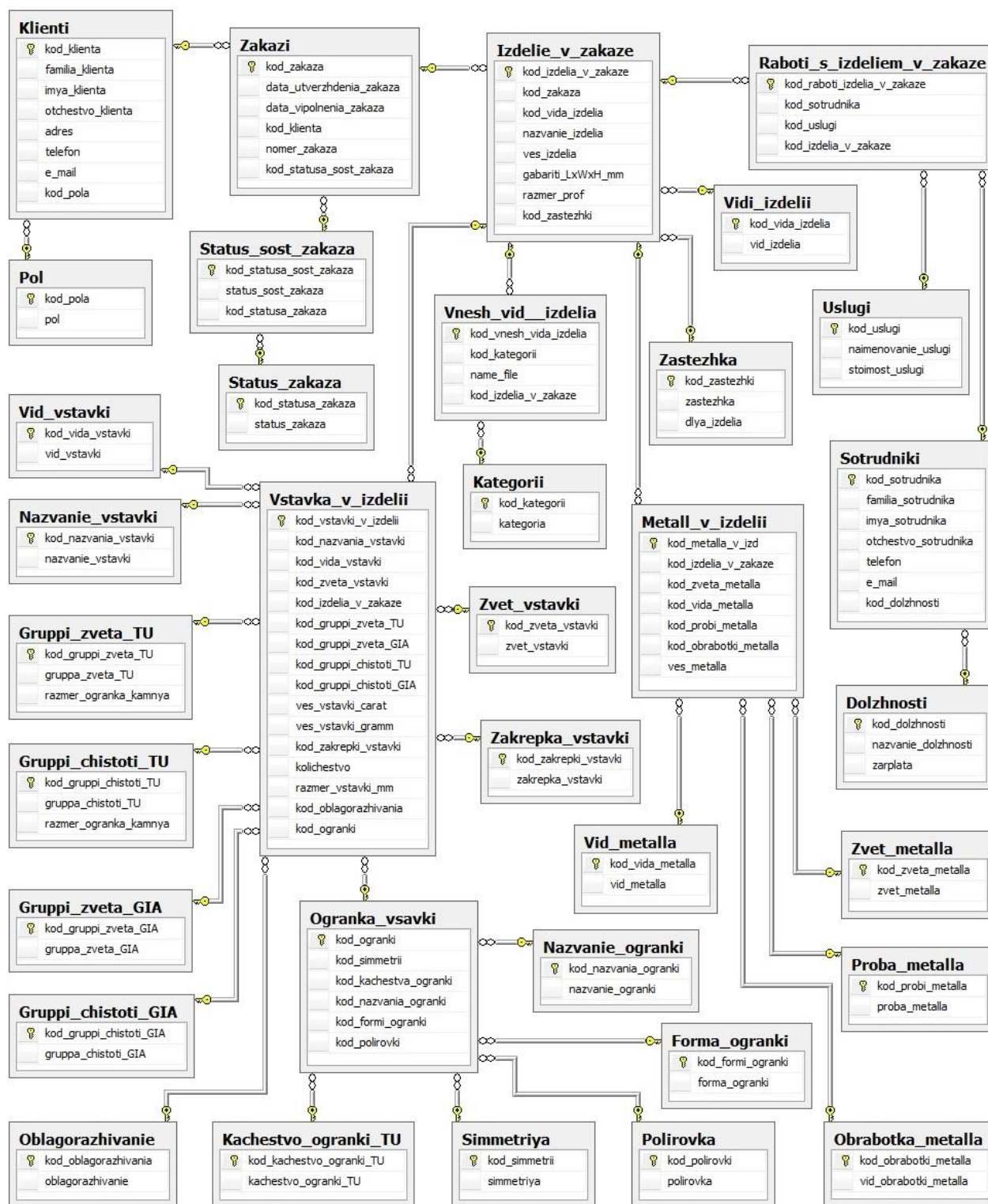


Рис. 2. Схема базы данных информационной системы для студии ювелирного дизайна

Очень важной частью базы данных должна стать систематизация всех используемых при подготовке и выполнении заказа материалов. К ним можно отнести: собираемые на стадии дизайна изображения-аналоги (по концепции, по видам изделий, материалам, по конструктивным особенностям, по стилю, форме или цвету и т.д.), которые можно сохранять в базе в виде коллекций, систематизировать и организовывать их удобный поиск. Это имеет смысл, потому что созданные коллекции могут быть применены многократно при работе с разными заказами. Весь объем предварительных эскизов также необходимо сохранять в базе. Отдельно стоит сказать и о том, что именно эскизы изделий, выполненные «от руки», как правило, не попадают ни в какие базы, а хранятся на бумажных носителях. Перевод ручных

эскизов в цифровой формат и сохранение их в базе, а также систематизация и сохранение всех результатов творческого поиска, происходящих на стадии дизайна изделия, должны стать правилом в организации деятельности студии. Результаты интеллектуального труда в наше время ценны как никогда и они должны пополнять имеющуюся базу данных и из информации преобразовываться в полезные знания.

Процесс постоянного накопления, анализа и получения новых данных будет способствовать повышению качества и скорости выполнения заказов, а также возможности принимать правильные решения в процессе управления заказами. Создание базы данных повысит эффективность деятельности дизайн-студии и будет способствовать поддержанию ее на конкурентоспособном уровне.

ВЫВОДЫ

Можно сделать вывод, что применение семантического подхода при разработке базы данных студии ювелирного дизайна позволяет в полной мере учесть особенности предметной области и специфику дизайнерской деятельности. Создание семантической модели процесса изготовления ювелирного изделия по индивидуальному заказу и, на ее основе, базы данных, позволит разработать специализированные информационные системы для организаций, деятельность которых связана с ювелирным дизайном, сделает возможным накопление данных об уникальных изделиях и расширение базы знаний о предметной области, что позволит анализировать и использовать полученные знания при работе с новыми заказами.

REFERENCES

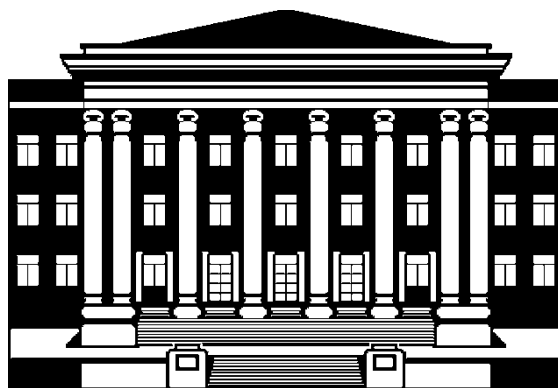
- [1] Бабанов, А.М. *Перспективы проектирования БД, открывающиеся с применением современных семантических моделей данных* // Вестник Томского Государственного Университета. Управление, вычислительная техника и информатика. – 2015. - №2 (31). – с.73-80
- [2] Бова, В.В. *Концептуальная модель представления знаний при построении интеллектуальных информационных систем* // Известия ЮФУ. Технические науки. Раздел III. Искусственный интеллект и нечеткие системы. – 2014. – с.109-117
- [3] Гаврилова, Т.А., Хорошевский В.Ф. *Базы знаний интеллектуальных систем: учеб. пособие для вузов* – СПб.: Питер, 2000. -384 с.
- [4] Голенков, В.В., Гулякина Н.А. *Проект открытой семантической технологии компонентного проектирования интеллектуальных систем. Часть 1: Принципы создания* // Онтология проектирования. – 2014. - №1(11). – С. 42-64.
- [5] Давыденко, И.Т. *Модель и средства компонентного проектирования баз знаний на основе унифицированных семантических сетей* // Электроника инфо. – 2013.- №11. – с. 23-25.
- [6] Дейт, К. Дж. *Введение в системы баз данных*. 8-е изд. Пер. с англ. М.: Вильямс, 2005. - 1328 с.
- [7] Кравченко, Ю.А. *Задачи семантического поиска, классификации, структуризации и интеграции информации в контексте проблем управления знаниями* // Известия ЮФУ. Технические науки. Раздел I. Анализ данных и управление знаниями. – 2016.- с.5-18
- [8] Кригель А., Трухнов Б. *SQL. Библия пользователя*, 2-е издание – Пер. с англ. – М.: Вильямс, 2010. – 752 с.
- [9] Олейник, А.Г., Ломов П.А. *Разработка онтологии интегрированного пространства знаний* / А.Г. Олейник, П.А. Ломов // Онтология проектирования. – 2016. – Т.6, №4 (22). – с.465-474
- [10] Сергиенко, М. А. *Особенности различных моделей представления знаний* / М.А. Сергиенко // Новая наука: стратегии и векторы развития. – 2017. – Т.2. - №2. – с.151-154

[11] Хомоненко, А.Д., Цыганков В.М., Мальцев М.Г. *Базы данных: учеб. для вузов* / В.М. Цыганков, М.Г. Мальцев; под ред. А.Д. Хомоненко. – 6-у изд., доп. – СПб.: КОРОНА-Век, 2011. – 736 с.

[12] *Руководство по SQL Server*. URL: <http://docs.microsoft.com/ru-ru/sql/sql-server/tutorials-for-sql-server-2016>

UNIVERSITY OF RUSE „ANGEL KANCHEV“

UNION OF SCIENTISTS - RUSE



**58-TH ANNUAL SCIENTIFIC CONFERENCE
OF UNIVERSITY OF RUSE „ANGEL KANCHEV“
AND UNION OF SCIENTISTS - RUSE**

24-25 OCTOBER 2019

I N V I T A T I O N

**Ruse, 8 Studentska str.
University of Ruse
Bulgaria**

PROCEEDINGS
Volume 57, Series 6.5.

Financial Mathematics
&
Informatics

Under the general editing of:
Assoc. Prof. Juri Kandilarov, PhD,
Slavi Georgiev

Editor of Volume 57:
Prof. Diana Antonova, PhD

Bulgarian Nationality
First Edition

Printing format: A5
Number of copies: on-line

ISSN 1311-3321 (print)
ISSN 2535-1028 (CD-ROM)
ISSN 2603-4123 (on-line)

The issue was included in the international ISSN database, available at <https://portal.issn.org/>.
The online edition is registered in the portal ROAD scientific resources online open access



PUBLISHING HOUSE
University of Ruse "Angel Kanchev"

<http://conf.uni-ruse.bg/>

РУСЕНСКИ УНИВЕРСИТЕТ "АНГЕЛ КЪНЧЕВ"
СЪЮЗ НА УЧЕНИТЕ – РУСЕ

- ENGLISH
- Начало
- Покана за участие
- Покана за участие (Филиал Разград)
- Покана за участие (Филиал Силистра)
- Покана за участие КАМ2018
- Гост лектори
- Програмнен комитет
- Работни езици
- Организационен комитет
- Тематични направления
- Изисквания към оформлението на докладите
- Етика за публикуване
- Такса за участие
- Срокове
- Програма на конференцията
- Публикуване на докладите
- Наградени доклади "Best Paper"
- Научни трудове на Русенски университет
- Адрес за кореспонденция

57-ма НАУЧНА КОНФЕРЕНЦИЯ
на Русенски университет „Ангел Кънчев“ и Съюз на учените – Русе
„Нови индустрии, дигитална икономика, общество – проекции на бъдещето“
25-26 октомври 2018 г.

НАУЧНА СЕСИЯ ЗА СТУДЕНТИ И ДОКТОРАНТИ

- Покана за участие
- Доклад-образец
- Сборници с доклади