UDC 004.021 DOI: 10.18372/2073-4751.77.18654

[1]**Zelenko E.V.**,
orcid.org/0000-0002-9939-3830,
e-mail: e.v.zelenko.fitis20@chdtu.edu.ua,
[2]**Kvasnikov V.P.**, Doctor of Technical Sciences
orcid.org/0000-0002-6525-9721,
e-mail: kvp@nau.edu.ua

MODELING AND ALGORITHMIZATION OF THE PRICING PROCESS IN ELECTRONIC COMMERCE USING SOFTWARE AGENTS

[1]Cherkasy State Technological University
[2]National Aviation University

Introduction

Improvements in the process of modeling and development of software tools in e-commerce using prototyping are caused by the growing demand for the development of complex computing systems in e-commerce business models and the development of an agent-oriented approach (agent-oriented programming, AOP) aimed to use such systems.

At the same time, there are a number of disadvantages in using certain approaches in e-commerce (namely dropshipping), which are not taken into account when developing software, which leads to delays in the delivery of goods and their sale at irrelevant prices.

Thus, there is a need to improve existing and create new methods of developing software used in e-commerce through the use of AOP, as well as a need to develop new methods of data processing in pricing when using dropshipping in e-commerce.

Purpose and Objectives

A significant contribution to the development of the theory and practice of using AOP in e-commerce was made by foreign and domestic scientists: Okumoku-Evroro O., Oseh V., Salmon I. A [1], Ramya S. Gowda [2], Luo X., Akkaladevi S. [3], Sproule S. [4], Berko A. Yu., Litvin V. V. [5], Vysotsky M. V., Guzhva V. M. [6], etc. However, the described approaches are not fully modeled or are formalized and are not universal when applying all e-commerce approaches, including the dropshipping.

Considering that the requirements for the design and development process of software used by companies at the initial stages of trading activities in the direct supply approach in e-commerce have their own specifics that influence the choice of approach and development tools, a relevant scientific task is aimed at researching and creating an approach to development of software tools used in the dropshipping (in e-commerce), which improves the characteristics of the software tool, as well as expanding its functionality and potential capabilities.

The purpose of the study is to improve the process of modeling and developing software using prototyping used in e-commerce, through the use of AOP to expand the variety of properties and potential capabilities of software, reducing the number of system processes used and connectivity between systems components, as well as increasing the speed of component development and the efficiency of processing input data in terms of calculation accuracy.

To achieve the purpose, the following tasks must be solved:

- conduct a theoretical study of the process of modeling software for an automated information system for updating retail prices, determine the number of system processes used, the connectivity between system components, the degree of expansion of many properties and potential capabilities of the software;
- conduct an experimental study of the software development process and determine the speed of software implementation of the user interface (UI) and asynchronous communication between autonomous system components;

- conduct an experimental study of the process of pricing algorithmization by a software agent and determine the effectiveness of the software implementation of the developed data processing algorithm in terms of calculation accuracy.

System requirements and AOP properties

The description of the functionality and limitations to the software (primary requirements) is presented by a high-level description of the task. The system must provide:

1. A permanent and automatic process of updating and forecasting retail prices, subject to the presence or absence of sufficient input data, respectively.
2. Detection of excessively inflated prices.
3. Increasing the speed of price updates without increasing the load on the system.
4. Expansion of functionality without increasing the load on the system.

4.1. The ability of the operator to interact with the system during the execution of the program to make decisions about excessively inflated prices.

4.2. The ability to interact between autonomous software modules performing different tasks to coordinate the use of rapidly changing data queues.

4.3. Ability to create and delete software for any department of the company without suspending the software of other departments.

5. Expanding development opportunities using artificial intelligence.
6. Simple and inexpensive user interface.
7. Ability to distribute the UI for use in different departments of the company.
8. The speed of development of the main functionality in the initial versions of the software and the possibility of further development with strict compliance with the requirements.
9. Generate output data after each price calculation process.

Let O be the finite set of the object, and A be the agent ($O \neq \emptyset$, $A \neq \emptyset$). $O \cup A = \{e, i, p, \lambda, \iota, \zeta, \mu, \delta, \epsilon, \eta, \kappa, \nu\} \cup \{f, r, s\}$, where λ is location, ι is identification, ζ is autonomicity, μ is inclusion, δ is mobility, ϵ is granularity, η is cognition, κ is flexibility, ν is narrowness (based on [7]), e is encapsulation, i is inheritance, p is polymorphism, f is expansion of a set of properties without increasing the load on the system, r is reactivity (response to environmental changes), s is speed of development. Then $O \subset A = \{e, i, p, \mu, \iota, \epsilon, \eta, \kappa, \nu\}$, $A \setminus O = \{\lambda, \zeta, \delta, f, r, s\}$.

It is necessary to satisfy the formalization of the task (primary requirements). Let R be the finite set of satisfied primary requirements of the customer ($R_n = \{1_1, 2_2, 3_3, 4_4, 4.1_5, 4.2_6, 4.3_7, 5_8, 6_9, 7_{10}, 8_{11}, 9_{12}\}$), where numbers indicate the corresponding elements of the list of requirements, and n – sequence number of the element in the tuple. Then: $f \to R_4$, $r \to R_5$, $\zeta \to \{R_6, R_7\}$, $s \to R_{11}$.

Since the set of properties of the application of AOP platforms satisfies 5/12 of the primary requirements (fig. 1), a decision was made to use it at the stage of software development in order to improve the software implementation (fig. 2).

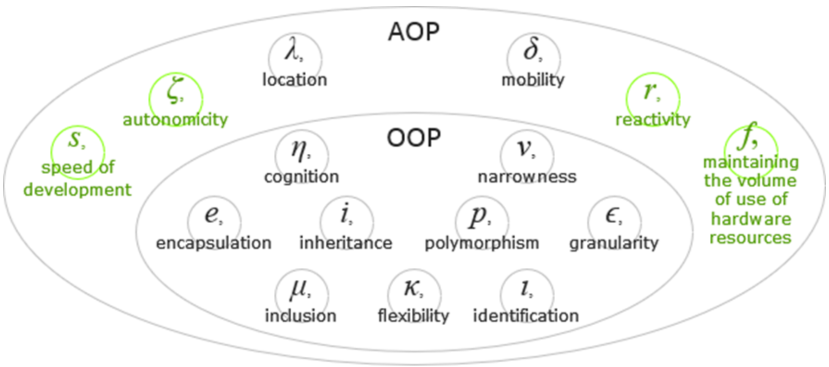

Fig. 1. Sets of OOP and AOP properties

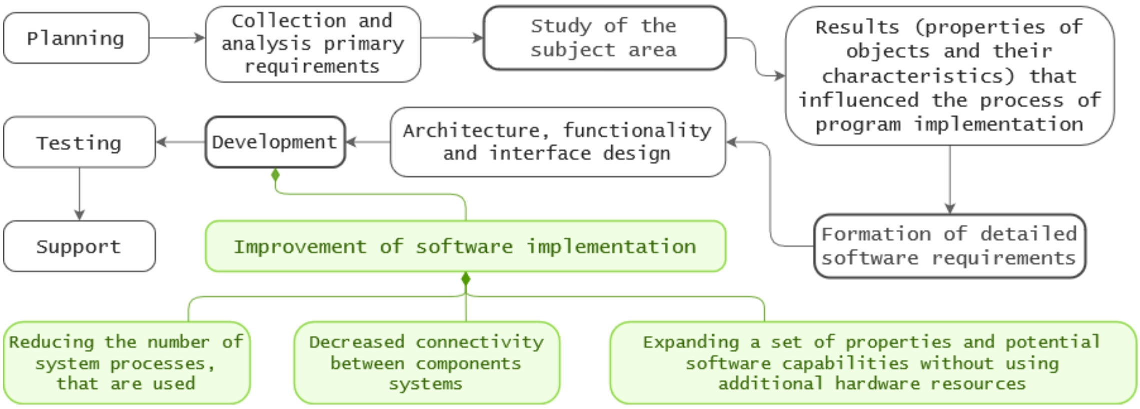

Fig. 2. An improved software development method based on the dropshipping in e-commerce using AOP

As a result of the improvement, the number of system processes was reduced by 4 units (fig. 3, 4) and the connectivity between system components by 1-3 connections (fig. 5, 6), as well as to expand the set of properties and potential capabilities of the software tool by 8 elements (fig. 7).

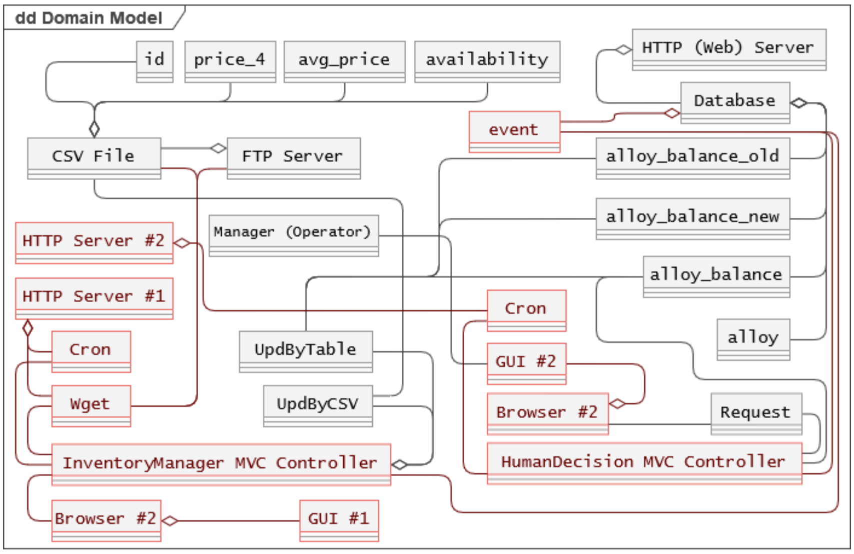

Fig. 3. Number of system processes before using an improved software development method (4 processes)

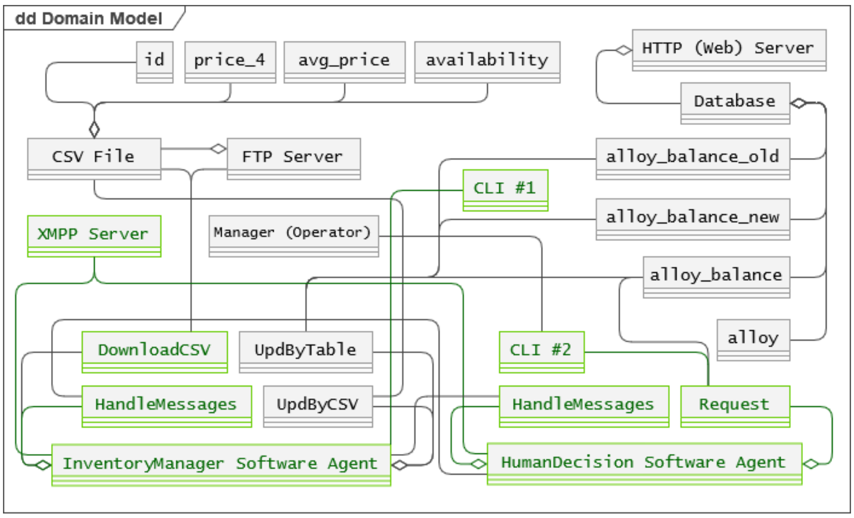

Fig. 4. The number of system processes after applying the improved method

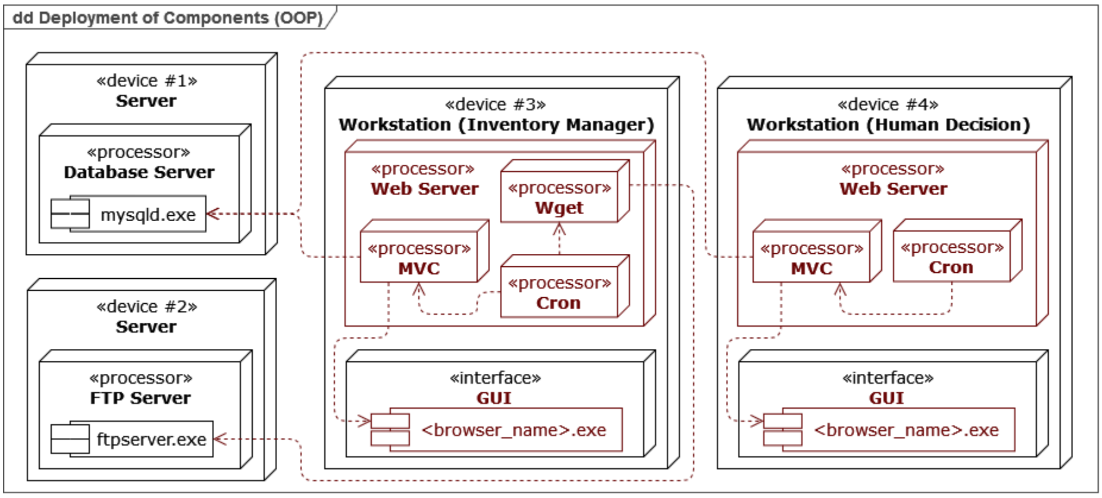

Fig. 5. The number of connections between system components before applying the improved method

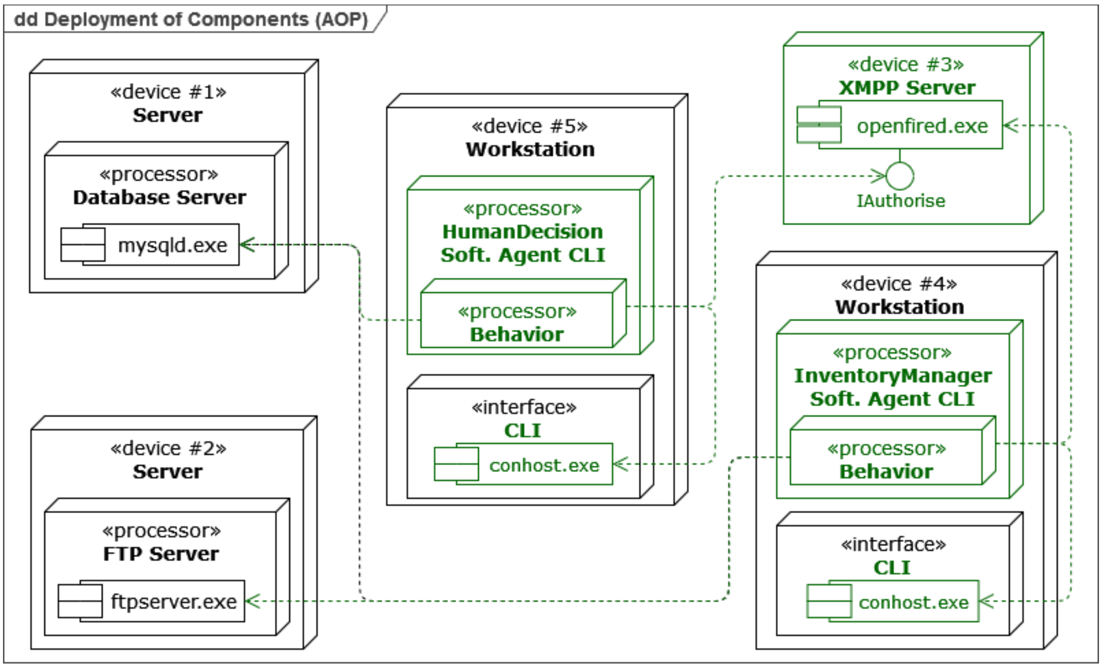

Fig. 6. Reducing the connectivity of software components after applying the improved method

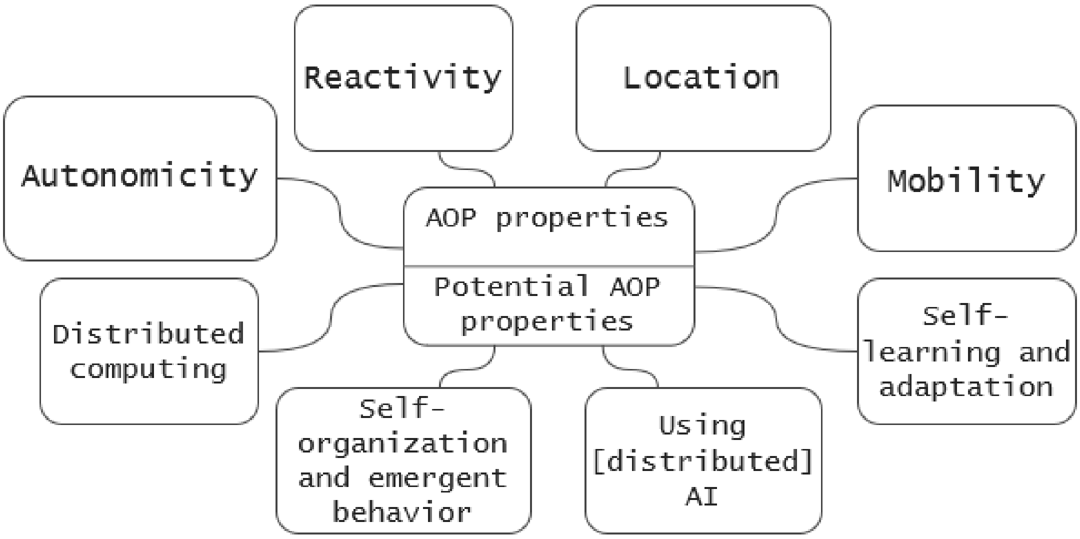

Fig. 7. The main properties and potential opportunities of AOP and software agents

Increasing the speed of UI development

Based on the results of a comparative analysis by the LOC metric of the amount of software code written using the traditional approach and the AOP platform, it was concluded that the software code according to the new method is shorter, and therefore requires less resources for implementation (true for automatic registration, messaging, using templates, etc.). Using the example of software implementation of asynchronous communication, this indicator is $\left|\left(\frac{24}{37}-1\right)\right| \cdot 100\,\% = 35{,}14\,\%$ (fig. 8, 9). It is assumed that the amount of code written can be an indicator of the productivity and efficiency of the development team, influencing the total time spent on the project.

```python
import asyncio
import aiohttp

async def send_message(): # Відправник
    msg = { # Створення повідомлення
        'to': 'receiver@xmpp_server',
        'body': 'Hello World'
    }
    async with aiohttp.ClientSession() as session:
        # Відправка повідомлення
        try:
            async with session.post('http://your_server/send_message', json=msg) as response:
                if response.status == 200:
                    print('Message sent successfully!')
                else:
                    print('Failed to send message!')
        except Exception as e:
            print(f'Exception: {e}')

...
async def receive_message(): # Отримувач
    while True:
        async with aiohttp.ClientSession() as session:
            async with session.get('http://your_server/receive_message') as response:
                if response.status == 200:
                    msg = await response.json()
                    print(f'Message received with content: {msg["body"]}')
                else:
                    print('Failed to receive message')

        await asyncio.sleep(10) # Очікування 10 с перед наступною перевіркою
...
async def main():
    # Створення завдань для відправки та отримання повідомлень
    send_task = asyncio.create_task(send_message())
    receive_task = asyncio.create_task(receive_message())
    # Очікування завершення завдань
    await asyncio.gather(send_task, receive_task)
```

Fig. 8. Software implementation of communication between software components without an AOP platform

```python
import spade
from spade.agent import Agent
from spade.behaviour import OneShotBehaviour
from spade.message import Message

class SenderAgent(Agent):
    class InformBehav(OneShotBehaviour):
        async def run(self):
            msg = Message(to='receiver@xmpp_server')    # Створення екземпляру повідомлення
            msg.body = 'Hello World'                    # Контент повідомлення
            try:
                await self.send(msg) # Відправка повідомлення
                print('Message sent successfully!')
            except Exception as e:
                print(f'Failed to send message: {e}')

...
class ReceiverAgent(Agent): # Отримувач
    class RecvBehav(OneShotBehaviour):
    async def run(self):
        msg = await self.receive(timeout=10) # Очікування 10 с на повідомлення
            if msg:
                print(f'Message received with content: {msg.body}')
            else:
                print('Did not received any message after 10 seconds.')
```

Fig. 9. Software implementation of communication between software components using the AOP platform

The recording and comparison of time indicators for UI development (including front-end business logic) according to traditional and new methods were also carried out. The analysis of the received data confirmed an increase in the speed of the software implementation process by $\left|\left(\frac{1552}{5059}-1\right)\right| \cdot 100\ \% = 69{,}32\ \%$ when using the improved method compared to the traditional one (Fig. 10) when reducing the code volume indicator by metric LOC by $\left|\left(\frac{89}{152}-1\right)\right| \cdot 100\ \% = 41{,}45\ \%$.

%	Time	Activity
64%	54m 23s	sublime_text
22%	18m 21s	Firefox
10%	8m 27s	dbforgemysql
3%	2m 35s	Total Commander
	18s	Notepad
	9s	shellexperiencehost
	6s	Windows Explorer

a.

%	Time	Activity
54%	14m 3s	sublime_text
23%	5m 55s	py
15%	3m 51s	Total Commander
6%	1m 27s	Firefox
1%	12s	Task Manager
1%	10s	openfire
	7s	Windows Explorer
	7s	dbforgemysql

b.

Fig. 10. Increasing the speed of UI development according to the improved method (based on the results of "RescueTime"): a – without the AOP platform: WUI, $time_{dev}$ = 5059 s (LOC = 152); b – with the AOP platform: CLI, $time_{dev}$ = 1552 s (LOC = 89)

Due to the use of the software agent development environment and, as a result, the absence of the need for GUI development, the time spent on UI development (including front-end business logic) is reduced (hence, labor intensive; compared to the development of such for web applications).

Modeling the process of processing input data by a software agent in pricing

To optimize data processing processes by software agents when applying the dropshipping in e-commerce, an experimental study of the pricing algorithmization process was conducted (fig. 11).

In order to find the optimal parameters of the pricing equation (1), a heuristic algorithm for the approximate solution of the problem was developed (similar to the SUMT method [8, p. 65-68]), which is based on the analysis of statistical data and the use of interpolation and the problem was applied parametric optimization (2).

$$f = \bigcup_{i=1}^{2} f_i,$$

$$f_1(p) = p + p \cdot l \cdot t\frac{lm}{b}, p > \mu, \qquad (1)$$

$$f_2(p) = p + p \cdot l \cdot t\frac{lm}{b} + a \cdot t(\mu - p), p \le \mu$$

$$\begin{cases} lm \in \{8, 12\} \\ b \in [1, 100] \\ p \in (0, \infty) \\ \mu \in (0, \infty) \\ t \in [1, 20] \end{cases}$$

where p is the price for 4 units of the product; lm is a constant with a value of 8/12 (depending on the season); b – balance (the balance of goods in the warehouse); μ – is the average price by suppliers for each unit; a – element of set A, or APC – average price coefficient; l is an element of the set L (or LBC) – the coefficient lm/b; t or TC is the time coefficient; the number of hours that have passed).

$$P_{\%}(l_i, a_j, \Delta\bar{p}_{\%}) \to \min \qquad (2)$$

where $P_{\%}$ – the percentage ratio of the number of new price values, which are higher than the calculated values, in relation to the total number; a and l – parameters located on the given segments A and L, provided that $\Delta\bar{p}_{\%} = 5$ (the growth rate of the arithmetic average of prices). It is necessary to find such values of the parameters l, a and $\Delta\bar{p}_{\%}$, that ensure the minimum of the objective function. The complete list of constraints-equalities is given in the form of system (3).

$$\begin{cases} 0{,}0003 \le l_i \le 0{,}003 \\ 0{,}01 \le a_j \le 0{,}1 \\ i,j \in [1,26] \\ \Delta\bar{p}_{\%} = 5 \end{cases} \quad (3)$$

After the generation and calculation of the values of the dependent parameters, the optimal values of the coefficients were determined using the minimum value $P_{\%min} \approx 11{,}6927$: $LBC_{opt} \approx 0{,}0016$ and $APC_{opt} \approx 0{,}0387$, where opt. is the optimal value. The values of the variables APC_{opt}, $P_{\%min}$, LBC_{opt} are expressed through coordinates X, Y, Z respectively, for the graphical solution of the optimization task (fig. 12).

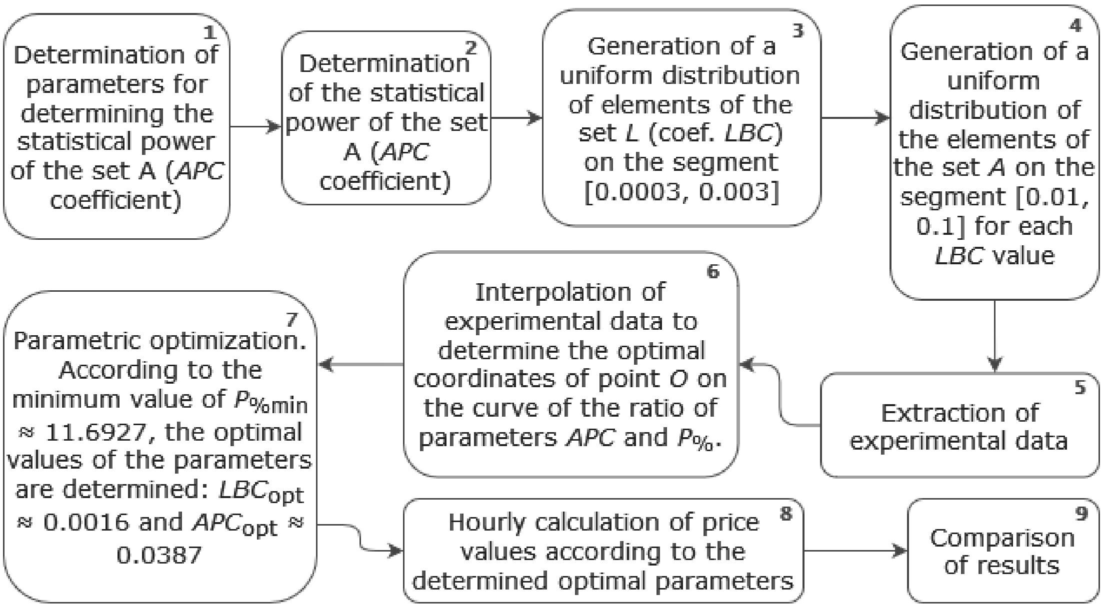

Fig. 11. The general scheme of the study

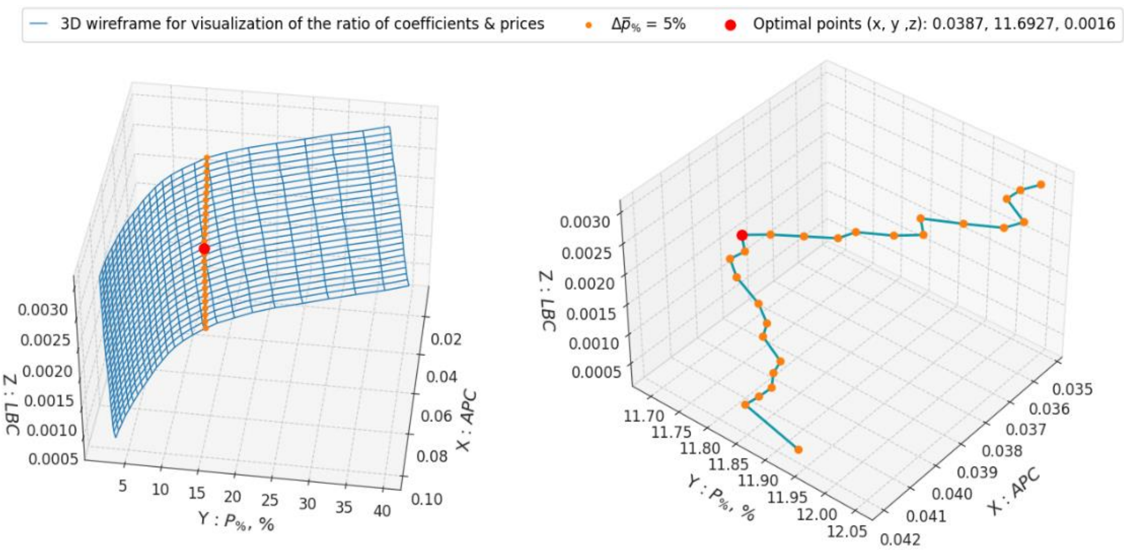

Fig. 12. 3D visualization of the ratio of coefficients and prices

Given that the algorithm for the approximate solution of the parametric optimization problem is heuristic, the obtained parameter values can be considered suboptimal. The values of the variables APC_{opt}, LBC_{opt}, *lm* (8) and *TC* (20) are substituted into the price update algorithm to generate a CSV file with calculated retail prices. The results after calculation are presented in fig. 13 and tab. 1.

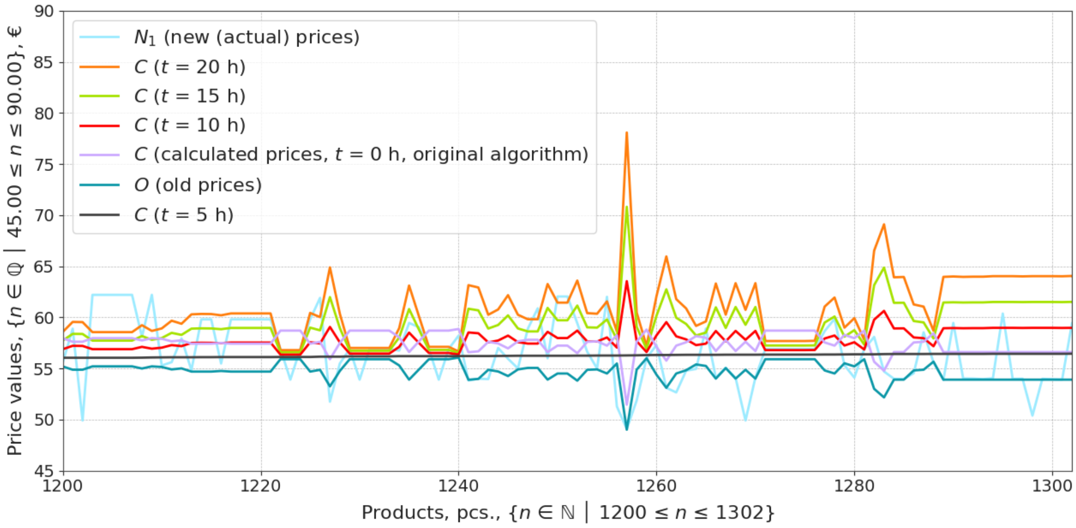

Fig. 13. Prices: old, calculated (original algorithm: t = 0 h; new algorithm: t = 5/10/15/20 h), actual (new) for: $1200 \leq x \leq 1302$, $45 \leq y \leq 90$

Table 1. Values of $\bar{p}$, $\Delta\bar{p}_{\%}$ and $P_{\%}$ for the results of each type of algorithm

#	Type	$\bar{p}$	$\Delta\bar{p}_{\%}$	$P_{\%}$
1	O	139,20	–1,24	60,85
2	N_1	140,94	–	–
3	C_{BM}	146,16	3,70	**22,81**
4	C_1	139,63	–0,92	48,47
5	C_2	140,07	–0,61	46,24
6	C_3	140,51	–0,30	44,15
7	C_4	140,95	0,01	41,87
8	C_5	141,39	0,32	39,66
9	C_6	141,83	0,63	37,80
10	C_7	142,27	0,95	35,83
11	C_8	142,71	1,26	33,64
12	C_9	143,15	1,57	31,27
13	C_{10}	143,59	1,88	29,02
14	C_{11}	144,03	2,19	27,07
15	C_{12}	144,46	2,50	25,05
16	C_{13}	144,90	2,81	22,98
17	C_{14}	145,34	3,13	20,79
18	C_{15}	145,78	3,44	18,89
19	C_{16}	146,22	3,75	17,25
20	C_{17}	146,66	4,06	15,54
21	C_{18}	147,10	4,37	14,03
22	C_{19}	147,54	4,68	12,79
23	C_{20}	**147,98**	5,00	**11,75**

The presence of a difference between the expected (5 %) and actual (3,7 %) values of the growth rate $\Delta\bar{p}_{\%}$ for the cost method of pricing was influenced by the absence of coefficients in the calculation, which is not an error, since the algorithmization of the pricing process was carried out during study of the algorithm with coefficients (C_{20} for t = 20 h), for which the actual value of the growth rate is equivalent to the expected one (5 %). The effectiveness of the new approach is determined by the growth rate $P_{\%}$ (the ratio of the number of new price values that are higher than the calculated values to the total number; $\Delta P_{\%\%}$), the value of which differs for the same values of the growth rate of the arithmetic average prices $\bar{p}$ ($\Delta\bar{p}_{\%}$). For this, the value of $P_{\%}$ between algorithm types C_{15} and C_{16} is determined using equation of interpolation:

$$x_1 = x_0 + (x_2 - x_0) \times \frac{f_2(x_1) - f_2(x_0)}{f_2(x_2) - f_2(x_0)} = 18{,}89 + (17{,}25 - 18{,}89) \times \frac{3{,}7 - 3{,}44}{3{,}75 - 3{,}44} \approx \mathbf{17{,}51}\ \%$$

Based on the reasons described above for the task of minimizing the value of $P_{\%}$, it can be argued that the lower the given value, the more effective the calculation. So, with the same value of the growth rate of the arithmetic average prices ($\Delta\bar{p}_{\%}$ = 3,7 %) the number of new price values that are higher than the calculated values is less by 22,81 % – 17,51 % = 5,3 %, that in terms of growth rate ($\Delta P_{\%\%}$) is **23,24 %** reducing the risk of loss on sale by a corresponding percentage, ceteris paribus.

Conclusion

In the course of the study, the actual scientific and applied problem of improving the process of modeling and development of software tools used in electronic commerce was solved by using an agent-oriented approach to expand the set of their properties and potential opportunities, reduce the number of used system processes and connections between components of the system, as well as increasing the speed of their development.

The main scientific and practical results obtained:

1. A theoretical study of the software modeling process of the automated information system of retail price update was conducted, during which the method of object-oriented programming for the development of software in electronic commerce was improved through the use of an agent-oriented approach, which made it possible to reduce the number of system processes that are used, by 4 units, to reduce the connectivity between system components by 1–3 connections to increase their autonomy, and also to expand the set of properties and potential capabilities of the software tool by 8 elements.
2. An experimental study of the software development process was carried out, during which the model of the implementation of evolutionary prototypes was improved due to the use of an agent-oriented approach to the development of software tools, which allowed to increase the speed of software implementation of the user interface by ~ 70% and asynchronous communication between autonomous system components on ~ 35%.
3. An experimental study of the process of algorithmization of pricing by a software agent was conducted, during which the efficiency of the software implementation of the developed data processing algorithm was increased in terms of calculation accuracy (by the rate of decline in the number of new price values that are higher than the calculated values) by 23.24% by improving the software implementation process pricing in electronic commerce using the method of object-oriented programming due to the study of data processing processes and the use of its results at the stage of requirements formation.

References

1. Okumoku-Evroro O., Oseh V., Salmon I. A. Use of software agents in e-commerce: benefits and applications. *International Journal of Advanced Engineering and Management Research*. 2017. Vol. 2, no. 1. P. 91–96. URL: https://www.ijaemr.com/view/1/2017.

2. Gowda Ramya S. Role of Software Agents in E-Commerce. *International Journal of Computational Engineering Research*. 2013. Vol. 3., no. 3. P. 246–251. URL: https://www.ijceronline.com/v3-i3.html.

3. Luo X., Akkaladevi S. Intelligent Software Agents Analysis in E-Commerce I. *Software Applications: Concepts, Methodologies, Tools, and Applications* / ed. by P. Tiako. Hershey : IGI Global, 2009. P. 1446–1451. DOI: 10.4018/978-1-60566-060-8.ch084.

4. Sproule S. Software agents in electronic commerce: a decision support systems approach : Ph.D. Thesis (Management Science / Systems). Hamilton, Canada, 2003. 234 p. URL: http://hdl.handle.net/11375/7540.

5. Берко А. Ю., Литвин В. В. Застосування інтелектуальних агентів в системах електронної комерції. *Інтелектуальні системи прийняття рішень і проблеми обчислювального інтелекту*, Залізний Порт, Україна, 21–27 травня 2018 р. Херсон, 2018. С. 145–147. URL: https://sci.ldubgd.edu.ua/bitstream/123456789/5157/3/isdmci_2018.pdf.

6. Висоцький М. В., Гужва В. М. Деякі аспекти мультиагентного моделювання систем електронного бізнесу (на прикладі інтернет-аукціону). *Вісник східноукраїнського національного університету імені Володимира Даля*. 2012. № 2(173). С. 173–178. URL: https://www.kpi.kharkov.ua/archive/articles/logistics/VISNYK_2_2012[1].pdf.

7. Zelenko E. Review of the mathematical model, properties, classes and other features of software agent development.

Electronic modeling. 2024. Vol. 46, no. 2. P. 3–14. DOI: 10.15407/emodel.46.02.003.

8. Методичні вказівки до лекційних зайнять з дисципліни «Чисельні методи в фізиці» для студентів спеціальності 153 «Мікро– та наносистемна техніка» / Уклад.: К. О. Мінакова та ін. Харків : НТУ «ХПІ», 2019. 73 с. URL: https://core.ac.uk/download/pdf/188178033.pdf

Zelenko E.V., Kvasnikov V.P.

MODELING AND ALGORITHMIZATION OF THE PRICING PROCESS IN ELECTRONIC COMMERCE USING SOFTWARE AGENTS

The article is devoted to the improvement of the object-oriented programming method for the development of software used in electronic commerce, using an agent-oriented approach.

Experimental studies were conducted to compare traditional and improved methods for developing software used in e-commerce. In the course of research due to the application of an agent-oriented approach and a platform for the development of software agents, the following were found: an increase in the speed of software implementation of the user interface and asynchronous communication between autonomous system components; reducing the number of system processes used; reduction of connectivity between system components; increasing the efficiency of the software implementation of the developed data processing algorithm in terms of calculation accuracy; expanding the set of properties and potential capabilities of the software tool.

Keywords*: software agent; agent-oriented programming; software; software modeling; user interface; software development.*

Зеленько Е.В., Кваснiков В.П.

МОДЕЛЮВАННЯ ТА АЛГОРИТМІЗАЦІЯ ПРОЦЕСУ ЦІНОУТВОРЕННЯ В ЕЛЕКТРОННІЙ КОМЕРЦІЇ ІЗ ЗАСТОСУВАННЯМ ПРОГРАМНИХ АГЕНТІВ

Стаття присвячена удосконаленню методу об’єктно-орієнтованого програмування до розробки програмного забезпечення, що використовується в електронній комерції, за допомогою застосування агентно-орієнтованого підходу.

Проведено експериментальні дослідження з метою порівняння традиційного та удосконаленого методу до розробки програмного забезпечення, що використовується в електронній комерції. В ході досліджень за рахунок застосування агентно-орієнтованого підходу та платформи для розробки програмних агентів виявлено: підвищення швидкості програмної реалізації інтерфейсу користувача та асинхронної комунікації між автономними компонентами системи; зменшення кількості системних процесів, що використовуються; зменшення зв’язності між компонентами системи; підвищення ефективності програмної реалізації розробленого алгоритму обробки даних за точністю розрахунку; розширення множини властивостей та потенційних можливостей програмного засобу.

Ключові слова*: програмний агент; агентно-орієнтоване програмування; програмне забезпечення; моделювання програмного забезпечення; інтерфейс користувача; розробка програмного забезпечення.*