

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Одеський національний технологічний університет
Університет Інформатики і прикладних знань, м.Лодзь, Польща
Національний технічний університет України «Київський
політехнічний інститут»
Навчально-науковий інститут комп'ютерних систем і технологій
«Індустрія 4.0» ім. П.М. Платонова

XXIII Всеукраїнська науково-технічна конференція
молодих вчених, аспірантів та студентів

«СТАН, ДОСЯГНЕННЯ ТА ПЕРСПЕКТИВИ
ІНФОРМАЦІЙНИХ СИСТЕМ І ТЕХНОЛОГІЙ»

Матеріали конференції



Одеса

20-21 квітня 2023 р.

Стан, досягнення та перспективи інформаційних систем і технологій / Матеріали XXIII Всеукраїнської науково-технічної конференції молодих вчених, аспірантів та студентів. Одеса, 20-21 квітня 2023 р. - Одеса, Видавництво ОНТУ, 2023 р. – 449 с.

Збірник включає матеріали доповідей учасників конференції, які об'єднані за тематичними напрямками конференції.

Збірник буде корисним як для фахівців і працівників фірм, зайнятих в області ІТ, так і для викладачів, магістрів і студентів вищих навчальних закладів, які навчаються за напрямками і спеціальностями програмного забезпечення, обчислювальної техніки і автоматизованих систем, прикладної математики та обробки інформації, буде корисним професіоналам з комп'ютерного моделювання та розробки комп'ютерних ігор.

Результати досліджень у збірнику представляють собою своєрідний зріз сучасного стану справ в перерахованих галузях знань, який може допомогти як фахівцям, так і студентам університетів скласти загальну картину розвитку інформаційних технологій та пов'язаних з ними питань.

Наукові праці згруповані за напрямками роботи конференції та наведені в алфавітному порядку прізвищ авторів.

Матеріали (тези доповідей) друкуються в авторській редакції. Відповідальність за якість та зміст публікацій несе автор.

Матеріали подано українською та англійською мовами.

Редактор збірника Котлик С.В.

майстрів виробничого навчання автотранспортного профілю в коледжах. Максимович О. (Глухівський національний педагогічний університет імені Олександра Довженка)	
24. Визначення тенденцій розвитку 3D-моделювання та їх вплив на професійну підготовку спеціалістів в цій галузі. Марчук М.С., Дубич К.П. (Рівненський державний гуманітарний університет)	160
25. Використання цифрового інструменту RadiAnt DICOM Viewer студентами III курсу спеціальності «Медицина» в практичних навчальних навичках вибіркового компоненту «Променева діагностика, променева терапія». Мацькевич В.М., Ленчук Т.Л. (Івано-Франківський національний медичний університет)	161
26. Інформаційна система навчально-наукових ресурсів кафедри. Мулик О.В., Свинчук О.В., Бандурка О.І. (Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»)	163
27. Організація комунікації деканату зі студентами. Камушков О.С., Нікітчина Т.М., Перетяка О.С. (Одеський національний технологічний університет)	164
28. ChatGPT in Undergraduate Education: Benefits, Concerns, and Future Prospects. Огарков А.В., Харитонова Л.В., Копка К.О. (National Transport University)	166
29. Методи навчання, практичного тренування та наукової діяльності. Поліщук В.О., Зінченко С.М., Матейчук В.М., Кириченко К.В. (Херсонська державна морська академія)	167
30. Використання ігрових технологій для вивчення іноземних мов. Рудницький М.І. (Національний Університет "Одеська Політехніка")	169
31. Розробка мобільного додатку для вивчення математики. Савченко С.Я., Вербинський Д.І, Мунтян І.В. (ВСП "Фаховий коледж промислової автоматики та інформаційних технологій ОНТУ)	171
32. Роль робототехніки та програмування в освіті майбутнього. Савчук В.А. (Національний університет «Одеська політехніка»)	173
33. Освітні напрямки навчання в сучасному світі. Ткаченко Л. А., Ющенко А. П. (Український державний університет імені Михайла Драгоманова)	175
34. Особливості україномовної термінології в галузі інформатики та комп'ютерних технологій. Франчук Р. В. (Український державний університет імені Михайла Драгоманова)	176
35. Телеграм бот для здобувачів вищої освіти. Черноусов Д.І., Бандурка О.І., Свинчук О.В. (Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»)	178
36. Модернізація навчальної платформи для ОНТУ. Стогул В.М., Болтач С.В., Корнієнко Ю.К. (Одеський національний технологічний університет)	180
Розділ 4: Проектування інформаційних систем та програмних комплексів	182
1. Innovation as an indicator of the success of the company's functioning on the example of the it holding "choco". Amirkhankyzy A., Kim E.R. (University "Turan", Kazakhstan)	182
2. Towards domain modeling approach to software development for bank information systems. Daas T.I. (V.N. Karazin Kharkiv National University)	183
3. Development of an intelligent decision support system for ship movement management considering ship operator fatigue. Nosov P.S., Koretsky O.A., Onyshko D.M., Makarchuk D.V. (Херсонська державна морська академія)	186
4. Digitalization of the loan/installment process on the example of the IT company globerce capital. Sergeev K.A., Kim Ye.R. (University "Turan", Kazakhstan)	188
5. Architecture of software for solving virtual promotion problem based on lambda approach. Orekhov S.V. (Національний технічний університет "Харківський політехнічний інститут")	189
6. Building a model based on NLP for the implementation of semantic search in the	191

the cooking time 10 minutes before the order is fully prepared, directs the courier to the restaurant using the “Delivery” application.

“Themis” can be said to be the queen of the entire “CHOCOFOOD” ecosystem, since all restaurant data and user data, order history, the entire information system is stored here. The opening of restaurants all over Kazakhstan, the whole system of bots, the appointment of orders by courier and all distribution goes on this system. Frontend development is based on the framework “Vue.js”.

When working in the Athena system to add a new partner restaurant, after its authorization, a field with an invitation code for adding a restaurant to the partner application will automatically appear. The database is automatically updated and the data is entered into the Themis system.” Also, this system provides for editing the restaurant menu, changing its work schedule, restaurant performance, etc.

The application “CHOCO Delivery” is intended for riders, which specifies the courier’s full name, phone number, active and completed orders, schedule and statistics.

A lot of business processes are automated through the use of chatbots, for example, transferring an order to a restaurant, providing feedback in the Intercom support chat, collecting customer information and speeding up problem solving with the help of easy-to-create custom chatbots without code.

Intercom is a communication platform for working with clients. It includes a set of integrated products for each team, including sales, marketing, products and support. Intercom mechanisms allow you to set up targeted communication with customers on the website, in the network and mobile applications, as well as by e-mail [2].

In addition, in the Salesbot designer, you can automate business processes in just a couple of minutes: for example, transfer a transaction to a certain stage, set a task to a manager or send a letter to a client.

Summing up, we can say that in the near future, in conditions of fierce competition, not only large, but also medium-sized and smaller companies will switch to business process automation and try to introduce innovative solutions into their processes. Observing the work of such a powerful ecosystem, I can say from personal experience that there will always be improvements and innovations.

References:

1. Mashkovtsev P.V., Yashin S.N. Performance indicators of the innovative process of the enterprise // Bulletin of Kazan Technological University. - 2011. – pp. 288-291.
2. <https://startpack.ru/application/intercom>

UDC 004.41.02

TOWARDS DOMAIN MODELING APPROACH TO SOFTWARE DEVELOPMENT FOR BANK INFORMATION SYSTEMS

DAAS T.I. (timurkadaas@gmail.com)
V.N. Karazin Kharkiv National University

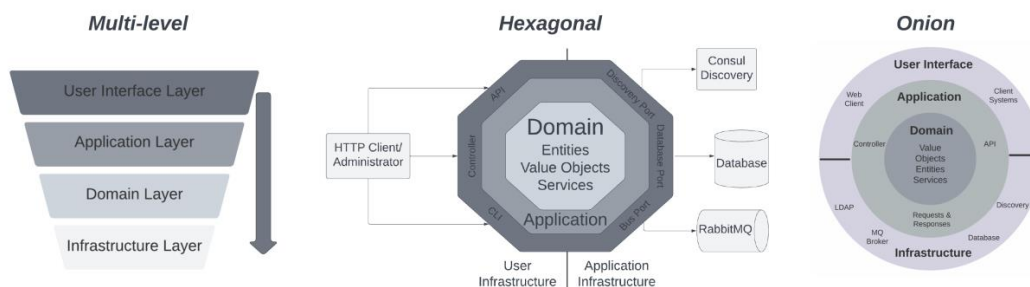
This paper considers main principles of domain models. The possibility of their use in the design and development of banking information systems was also analyzed.

At the current stage of the development of information technologies, domain modeling is becoming an increasingly used style in the construction of complex distributed information systems.

The main approach is Domain Driven Design (DDD), which was introduced in the early 2000s. It is designed to simplify the interaction and cooperation of domain experts and software developers at all stages of the product life cycle. DDD is an approach to software development that divides development into domain areas, which in turn should be designed and developed as independently as possible from each other. The main principle of DDD is bounded context, which consists in clearly defining the boundaries of the model's use [1].

One of the advantages of DDD is that there is no need to use specific architectures. This allows you to build systems using a monolithic architecture (MA) or using microservices(MSA). And if the advantages of the MA include the simplicity and speed of development, especially in the initial phases, then the disadvantages of this solution are the cohesion of the code, the complexity of horizontal scaling and the concentration of all bounded contexts in one place without any physical boundaries between them, which is a direct danger to the preservation DDD approach. In turn, the MSA does not have the listed disadvantages, and more importantly, it is itself based on the principles of a bounded context, when each microservice implements only certain dedicated business logic and nothing more.

The most popular architectural patterns of DDD are the Multi-level, Hexagon and Onion architectures. The features of each of the patterns are shown in picture 1.



Picture 1 – DDD architecture patterns

All these patterns are easily combined on the principles of MSA. When implementing an information system based on MSA, the Infrastructure level can include: Discovery Service, Config Service, Gateway, Load Balancer, Database, MQ Broker and other services. The Application level will include the used framework, which provides MVC controllers and tools for processing various input data formats which are implemented in the system. The domain level is presented in the form of services, repositories, objects and aggregators [2].

When migrating from MA to MSA, the DDD principle can be used to decompose the system into services, especially for complex distributed systems that require the allocation of bounded contexts. At this time, you also need to choose the basic principle of data exchange between contexts. There are two main principles: direct service call and asynchronous messaging, for example via MQ Broker. At the same time, the level of the domain in each microservice should not depend on the chosen principle of data exchange, since this exchange should be performed at the level of infrastructure (for the sender) and application (for receiver) [3].

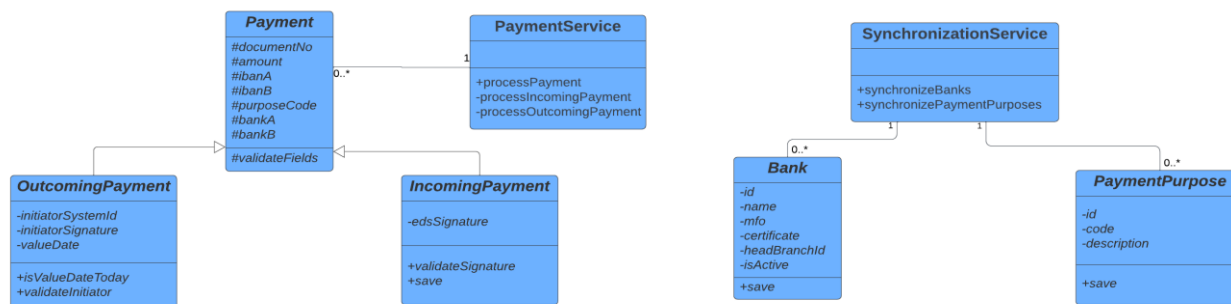
Banking information systems (BIS) differ in their specific requirements. Also, a feature of the banking sphere is that it unites BIS, which are quite different in purpose. This is due to the fact that banks actively interact both with clients and with each other. And also have complex internal processes that require the implementation of complex distributed high-performance systems. To meet these requirements and simplify the software development life cycle (SDLC), it is expedient to use a model approach, in particular DDD [4].

Let's consider the features of using the DDD approach on the example of two systems: the module of interaction with the system of electronic interbank payments (SEP) and Internet banking.

The software for integration with the SEP is a system in which two main domains can be distinguished: the exchange of payment instructions for proceeding users' payments between payer

and correspondent and the synchronization of reference information to keep up to date system metainformation. Moreover, the first domain can be divided into two subdomains: incoming payments and outgoing payments. UML class diagram for these domains is shown in Picture 2.

Much more complex from the point of view of building domains is the internet banking system, which must implement all functions for the client's interaction with the bank. Domains whose boundaries can be clearly defined are: users and their rights, authentication in the system, confirming transactions and actions in the system (electronic signature, OTR codes, etc.) and working with personal and system settings.



Picture 2 – UML class diagram for SEP system

Such domains have a bounded context and can be quite simply divided into separate microservices. But at the same time, there are certain domains that, due to their essential connection, are quite difficult to separate within contexts, which in turn causes difficulties when dividing into microservices and organizing data exchange procedures between microservices. Examples of such domains are payments and document exchange. First, payments are divided by types, the processing of which can be different, which makes it necessary to separate each type into a separate domain and, accordingly, a microservice. And secondly, certain types of payments require the attachment of various documents, the processing of which in turn affects the process of processing the payment itself. But since the domain of document exchange is a separate bounded context, it must be moved into a separate microservice, so in the domain of a specific type of payment, "compensatory measures" should be provided at the service level.

This work examines the advantages and features of using domain models in the designing BIS. The proposed approach will be used in the course of further research. It is planned to develop a domain model for one of the BISs, analyze its attributes and prove the effectiveness of using this approach during designing and developing BISs.

REFERENCES

1. V. Vernon, *Implementing Domain-Driven Design*. Boston, USA: Addison-Wesley, 2013
2. V. Nair, *Practical Domain-Driven Design in Enterprise Java*. New York, USA: Apress, 2019
3. S. Millett, *Patterns, Principles and Practices of Domain-Driven Design*. Hoboken, USA: John Wiley & Sons, 2015
4. J. Korsun, *5 Innovations in Banking Application Development: From Challenges to Opportunities*, 28.06.2019. [Online]. Available: <https://djangostars.com/blog/banking-application-development>. Accessed on: 12.04.2023.