CHAPTER
SYSTEM-ANALYTICAL MODELING OF THE OPERATIONAL ACTIVITIES OF A BANK BRANCH BASED ON SIMULATION AND INFORMATION TECHNOLOGIES

DOI: https://doi.org/10.30525/978-9934-26-690-4-10

Iryna Danylyuk
PhD in Economics, Associate Professor,
Associate Professor at the Department
of Economic Cybernetics and Informatics
West Ukrainian National University

Summary

The study explores theoretical, methodological, and applied aspects of simulation modeling of banking systems in the context of digital transformation. The feasibility of using simulation modeling as a tool of systems analysis for complex socio-economic systems-characterized by stochasticity, dynamism, and nonlinear behavior-is substantiated. The specific features of applying the main modeling paradigms-discrete-event, agent-based, and system dynamics-are revealed in the context of analyzing banking business processes. The impact of modern information technologies and service-oriented architectures on the organization and functioning of banking systems is examined. The study decomposes the operational activities of a bank branch and identifies key business processes that shape the workload of the customer service system. Based on systems analysis and the use of Computer-Aired Systems Engineering, a conceptual model of a bank branch as a multichannel queueing system is developed. A formalization of input parameters, resources, and performance indicators is proposed, enabling the conduct of simulation experiments. The choice of the modeling environment is justified, and the expediency of employing hybrid approaches to reproduce both processual and behavioral aspects of system functioning is demonstrated. The obtained results provide a methodological foundation for developing a software-based simulation model to support managerial decision-making in the banking sector.

Introduction

The modern banking sector operates under conditions of high dynamism, digital transformation, and considerable uncertainty, which requires bank management to adopt flexible and effective managerial decisions. Traditional analytical methods are often unable to fully account for the complexity of interconnections, the stochastic nature of service demand, and the nonlinear dynamics of banking systems.

In this context, simulation modeling emerges as one of the most powerful tools of systems analysis for the study of complex systems [1]. It enables the reproduction of an object's functioning over time, allowing experimentation with the model, analysis of various "what-if" scenarios, and forecasting of system behavior without risks to the actual operations of the bank.

Simulation modeling becomes particularly relevant for the evaluation and management of key banking processes, such as financial stability [2] or the optimization of credit practices under crisis conditions [3]. This approach not only helps to identify bottlenecks in queueing systems or risk management but also provides a means to assess the effectiveness of implementing new digital tools and business processes.

It should be emphasized that simulation modeling has deep historical roots in the disciplines of operations research and systems analysis. Initially employed in military and industrial contexts, it gradually expanded its scope to encompass economics, finance, and management. This evolution is attributable to the inherent complexity and uncertainty of socio-economic systems, particularly those within the banking sector, where multifactorial interactions render traditional forecasting methods insufficiently effective.

Within the banking domain, simulation models provide management with the capacity not only to examine the current state of the system but also to anticipate the implications of strategic decisions. For instance, such models can facilitate the identification of optimal liquidity management policies, the assessment of regulatory changes, and the projection of outcomes associated with the introduction of new financial products. In this regard, simulation modeling functions as a decision-support mechanism, mitigating risks and enhancing managerial effectiveness.

In the contemporary era of banking sector transformation, the relevance of simulation modeling has intensified. The integration of Big Data, machine learning algorithms, and artificial intelligence enables the construction of more precise and adaptive models. These advances allow banks to respond more rapidly to market fluctuations, forecast customer behavior with greater accuracy, and streamline internal business processes. Consequently, simulation modeling emerges not merely as an analytical instrument but as a pivotal component of the digital strategies underpinning the development of banking institutions.

1. Theoretical and methodological foundations of simulation modeling of banking business processes under digital transformation

Simulation modeling is one of the most effective tools of systems analysis for the study of complex systems. It provides the ability not only to formally describe an object but also to reproduce its dynamic functioning over time. This makes it possible to conduct experimental studies of the model, evaluate different development scenarios (“what-if”), and forecast system behavior without risks to the actual operations of a banking institution [4].

Systems analysis (SA) is a key methodology of scientific inquiry that considers research objects as integrated systems. It represents a set of methods and a sequence of actions aimed at establishing structural relationships among the elements of the system under study, with the goal of optimizing its functioning and supporting well-grounded managerial decisions. One of the central instruments in the arsenal of systems analysis is modeling, which enables the study of the properties of the original object by examining its simplified copy-the model.

Within systems analysis, two fundamental approaches to modeling are traditionally distinguished:

1. Analytical modeling (for example, methods of queuing theory, linear programming) is aimed at obtaining an exact, “closed-form” mathematical solution (e.g., in the form of formulas) that describes the state of the system. Its advantage lies in the speed of calculations and the ability to prove the optimality of the solution. However, this advantage is achieved at the cost of significant simplifications and assumptions (such as the requirement of process stationarity, Poisson arrival streams, or exponential service-time distributions), which are often not met in real complex systems.
2. Simulation modeling, in contrast to analytical modeling, does not seek an exact closed-form solution. By definition, simulation modeling is the process of designing a computerized model of a real system and conducting experiments with this model in order to understand its behavior or evaluate different management strategies.

The essence of simulation modeling lies in reproducing (simulating) the functioning of an object over time, with the system’s state changing either discretely or continuously under the influence of various events. Instead of obtaining an analytical solution, the researcher conducts a series of computational experiments with the model, collecting statistical data on its behavior, which are then analyzed.

For the study of banking systems, the advantages of simulation modeling are decisive due to its fundamental properties:

1. Stochasticity-demand for services, customer arrival times, duration of operations, and events are random in nature, which is difficult to describe using simple analytical distributions.
2. Nonlinear system behavior-for example, an increase in queue length may nonlinearly raise the probability of customer abandonment; operator fatigue may reduce work speed under peak load conditions, and so forth.
3. Dynamism-the system undergoes constant changes over time; its parameters (such as customer flow intensity) are not stable throughout the working day.

Simulation modeling makes it possible to account for all these factors, and its primary value for bank management lies in the ability to create a "virtual laboratory" for safely testing managerial hypotheses. Bank executives can experiment with changes in the number of operators, the introduction of a new electronic queue management system, or modifications to the business process of loan application processing, analyzing various "what-if" scenarios. This enables quantitative assessment of decision outcomes (for example, how the average customer waiting time or staff workload will change) before their implementation in real operations, which directly contributes to enhancing the effectiveness of managerial decisions.

Thus, within the framework of systems analysis, simulation modeling serves as a principal tool for studying complex socio-economic systems such as banks, allowing a transition from static analysis to dynamic experimentation and the identification of bottlenecks.

Simulation modeling is not a single monolithic method but rather a set of several fundamental paradigms. The choice of a specific paradigm depends on the level of abstraction at which the system is considered and the nature of the tasks to be addressed. Historically, three main approaches (or types) of simulation modeling have emerged and gained the widest application:

- System Dynamics (SD);
- Discrete-Event Simulation (DES);
- Agent-Based Modeling (ABM).

System Dynamics (SD) is a modeling paradigm initiated by Jay Forrester in the 1950s, used to understand the complex behavior of systems over time. It focuses on the macro level, viewing the system as a set of accumulators (stocks or aggregates), flows between them, and feedback loops that regulate these flows.

The essence of SD lies in identifying cause-and-effect relationships and the global influence of certain parameters on others. SD models are typically described in the form of systems of differential or difference equations.

System Dynamics (SD) is ideally suited for solving strategic tasks at a high level of abstraction. For example, SD can be used to model the overall financial stability of a bank, analyze the impact of macroeconomic factors (such as the National Bank of Ukraine's discount rate) on the size of the loan portfolio, or study the dynamics of competition among banks in the deposit market.

Discrete-Event Simulation (DES) is a meso-level, process-oriented approach employed when the functioning of a system can be reliably represented as a sequence of operations or processes.

The key feature of Discrete-Event Simulation (DES) is that the state of the model changes not continuously but only at discrete points in time when events occur (for example: a customer arrives at the branch, a customer joins the queue, an operator completes a service). Modeling in DES describes the movement of entities-such as customers, applications, or transactions-through the system, where they compete for resources-such as operators, ATMs, or servers-and, if resources are occupied, wait in queues.

DES is specifically used for modeling queueing systems. Tasks such as "customer service," "queue management," "resource management," and "bottleneck identification" are classical examples of the application of discrete-event simulation.

Agent-Based Modeling (ABM) is a relatively new approach (compared to SD and DES) that operates at the micro level and implements a "bottom-up" principle. The foundation of ABM is agents-autonomous, active entities that possess an individual state, a set of behavioral rules, and the ability to interact with one another and with the surrounding environment. Instead of describing global flows (as in SD) or processes (as in DES), ABM focuses on individual behavior.

The essence of ABM lies in the study of emergent behavior-complex global phenomena that arise from simple individual actions and interactions of agents. ABM enables the modeling of complex, adaptive behavior of customers or market participants. For example, in a bank branch model, a customer (agent) may independently make decisions: choose the shortest queue, switch to an ATM if operator queues are too long, or leave the branch altogether (so-called *balking* and *reneging*). At a higher level, ABM can model the transformation of lending practices under crisis conditions, where both banks and borrowers (agents) change their behavior.

To facilitate better understanding, a comparative characterization of the three key paradigms of simulation modeling is presented in Table 1.

Table 1

Comparative analysis of simulation paradigms

Characteristic	System Dynamics (SD)	Discrete Event Simulation (DES)	Agent-Based Modeling (ABM)
Abstraction Level	Macro-level (strategic)	Meso-level (process-oriented, operational)	Micro-level (individual)
Key Elements	Stocks, flows, feedback loops	Entities, resources, queues, processes	Agents, states, behavioral rules, environment
Time Progression	Mostly continuous	Discrete (event-based)	Mostly discrete
Key Question	How does the feedback structure influence global dynamics?	How to optimize the process, where are the "bottlenecks" and what is the throughput?	What complex system behavior emerges from individual interactions?
Example in Banking	Modeling a bank's financial stability; credit portfolio dynamics.	Modeling branch operations (queues, tellers); call center optimization	Modeling customer behavior in queues; modeling bank runs

Source: author's own

Modern software tools (for example, *AnyLogic*) also allow the creation of hybrid models, combining these paradigms.

The development of an adequate simulation model of a banking system is impossible without a clear understanding of its real architecture. A modern bank is, above all, a complex, high-technology information system. The processes of *digital transformation* and the implementation of *new digital tools* have fundamentally changed approaches to the design of banking IT systems.

Historically, banking systems evolved from monolithic architectures-large, centralized, tightly coupled systems (often running on mainframes), where all functions (deposits, loans, payments) were implemented within a single software complex. Such systems are difficult to modify and scale.

They were later replaced by modern distributed architectures, among which the following dominate:

1. Service-Oriented Architecture (SOA)-an approach in which the IT system is built as a set of independent, loosely coupled services that interact with each other through standardized protocols. Each service implements a specific business function (for example, “customer authorization service,” “account balance verification service,” or “interest calculation service”).
2. Microservices Architecture-in this approach, the system is divided into numerous small, fully autonomous services, each responsible for a very narrow business task. Every microservice has its own database and can be developed and deployed independently of the others.

The transition from monolithic to service-based architectures is of fundamental importance for simulation modeling. A banking business process (for example, “payment processing” or “loan issuance”) in modern architecture is no longer a single monolithic operation performed by one operator. Instead, it is a complex chain of sequential or parallel calls to various digital services, including: customer verification in the database, request to the anti-fraud system, account balance check, amount blocking, transaction submission to the payment system (such as SWIFT or the NBU SEP), receipt of confirmation, and balance update.

This generates a new class of problems that must be considered in the simulation model. Bottlenecks in such a system may be not only physical (an occupied operator, a queue at the cashier) but also digital (slow API response time of one microservice, limits on the number of simultaneous database connections, or delays in an external payment system).

To correctly model such a complex system, its preliminary formalization is required. Analysis of the subject area made it possible to decompose banking activities using standardized CASE tools and notations [5]:

– IDEF0-used for functional modeling of the “as-is” state, showing what the system does, its inputs, outputs, resources, and controlling influences.

– DFD (Data Flow Diagrams)-allows visualization of information flows between processes, data stores, and external entities (such as customers).

– UML (Unified Modeling Language)-the most widespread standard for describing IT systems. For modeling banking processes, the following UML diagrams are of particular value:

– Use Case Diagrams-describe the interaction of users (customers, operators) with the system.

– Activity Diagrams-provide a detailed description of the logic and sequence of steps within a business process.

Thus, the development of a simulation model is based on the results of business process formalization, which reflect the modern service-oriented architecture of banking IT systems.

For the practical implementation of simulation models based on theoretical paradigms and describing complex IT systems, a wide range of specialized information technologies and software tools is employed. The choice of tool is one of the key stages of modeling, as it determines the capabilities, flexibility, and speed of model development.

The software market for simulation modeling includes both narrowly specialized tools and universal platforms. Let us consider the key ones:

1. Arena (Rockwell Automation). This is one of the “classic” and most powerful tools specializing in discrete-event simulation (DES). Arena is widely used in industry, logistics, and services for process modeling, flow optimization, and analysis of queueing systems. Its advantage lies in the deep elaboration of the process-oriented approach [6].

2. Simulink (part of MATLAB). This tool is the de facto standard in engineering and scientific communities for modeling dynamic systems. Its strength lies in modeling continuous processes and implementing the system dynamics (SD) paradigm. In the banking sector, it is a powerful instrument for building macro-models (for example, modeling financial stability or risk management), but it is less intuitive and convenient for modeling operational processes based on queues (DES) [7].

3. AnyLogic. This software product natively supports all three major paradigms (SD, DES, and ABM) and allows them to be combined (hybridized) within a single model. This makes AnyLogic an exceptionally powerful tool for modeling complex socio-economic systems such as banks. For example, a researcher can:

– Use DES to model the service process in a branch (operators, cash desks, queues).

– Use ABM to model customer behavior (choosing a queue, going to a self-service terminal, or leaving the bank).

– Use SD to model the impact of strategic decisions (for example, an advertising campaign) on the overall customer flow in the bank.

AnyLogic provides powerful visualization capabilities and built-in libraries (for example, for pedestrian dynamics), which simplify the construction of branch models.

4. Python (with the SimPy library). This represents a fundamentally different approach, where instead of a visual constructor, the researcher uses a general-purpose programming language (Python) together with a specialized library (such as SimPy), which implements the logic of discrete-event simulation.

The formulation of a simulation modeling task is essentially a process of formalization and algorithmization. In general, any simulation modeling project includes the following key stages:

– Problem formulation and task definition-determining the goal, object, and subject of the study.

– Development of a conceptual model-formalizing the system, defining its boundaries, key elements, parameters, and logic of interaction.

– Development of a software (simulation) model-implementing the conceptual model in the chosen software environment.

– Assessment of adequacy and validation of the model-verifying whether the model corresponds to the real object.

– Planning and conducting experiments-designing an experimental plan for testing hypotheses and "what-if" scenarios.

– Analysis and interpretation of results-formulating conclusions and recommendations.

Since the key objects of modeling in our study are the processes of "customer service," "queue management," and "resource management," the central element of the task formulation methodology is the decomposition of the banking system in terms of queueing theory (QT).

Thus, a detailed methodology for developing a conceptual model of banking operations must include the formalization of the following components of a queueing system (QS):

1. Identification and description of the input stream. It is necessary to formalize the flow of customers (or digital transactions) arriving for service, focusing on the following parameters:

– Distribution law of interarrival times. It is critically important to move beyond the simplified assumption of a Poisson stream (the simplest flow), since the real customer flow is non-stationary (intensity differs in the morning, during the lunchtime peak, and in the evening) and cyclical (varies on salary/pension payment days compared to regular days).

– Source of the stream (finite or infinite).

– Heterogeneity (typification) of the stream. The flow must be divided into types (individual customers, corporate clients, VIP clients), as they may require different forms of service.

2. Description of the queue mechanism and discipline. This involves formalizing the rules by which customers wait for service. The following parameters must be specified:

– Queue capacity-either unlimited (theoretically) or limited (by the size of the physical premises or the buffer in an IT system).

– Queue discipline-defining the order in which customers are selected from the queue. This may not only be FIFO ("First-In, First-Out") but also priority disciplines (e.g., VIP customers served out of turn).

– Queue behavior-the methodology must account for behavioral aspects (which can be easily implemented in ABM or DES): *balking* (a customer refuses to join the queue if it is perceived as too long) and *reneging* (a customer leaves the queue after waiting for a certain period of time).

3. Description of the service mechanism. This involves formalizing the actual process of delivering banking services. The following parameters must be specified:

– Number of service channels-the number of parallel "servers" (operators, cash desks, ATMs, server streams).

– Structure of service-channels may be parallel and identical (any operator performs any operation), specialized (separate cash desk, separate loan manager), or sequential (multi-stage service).

– Distribution law of service time-service time must depend on the type of operation (utility bill payment: 1–2 minutes; deposit arrangement: 10–15 minutes; loan application review: 30–60 minutes).

4. Definition of output parameters and Key Performance Indicators (KPIs). At the final stage of task formulation, it is necessary to clearly define which metrics the model should generate in order to achieve the objectives of the study (improving decision-making efficiency). These include: average customer waiting time in the queue; maximum queue length; average time a customer spends in the system; staff (resource) utilization rate; percentage of "lost" customers (those who leave due to reneging or balking).

This methodology, based on the decomposition of banking operations in terms of queueing systems, makes it possible to structure a complex subject area.

2. Information and technological support for system-analytical modeling of banking branch operations

The banking system, as an object of system analysis, is characterized by a number of attributes that define its complexity: high dynamism, the stochastic nature of demand for services, and the nonlinear dynamics of functioning. To construct an adequate model, it is first necessary to decompose its structure.

Bank operational activities are divided into two key zones:

1. Front Office-units that directly interact with customers. In the context of the proposed model, focused on customer service in a branch, the front office includes tellers, cashiers, and loan managers. It is here that queues arise and services are directly provided.
2. Back Office-the set of units that ensure processing, verification, accounting, and support of operations initiated by the front office. This includes processing centers, risk management departments, IT divisions, and accounting.

For the purposes of simulation modeling, the research focus shifts specifically to the front office. Back-office processes are considered as external systems or sources of time delays (for example, the time required for loan approval) relative to the queueing model of the branch.

The main functions of the front office, which are the subject of modeling, represent classical queueing systems: the input stream consists of bank customers with different needs; the queue is the physical or virtual waiting of customers for service; the service channels are the bank employees (resources), such as tellers and managers.

Analysis of this structure confirms that it is insufficient to model only information flows (as in DFD) or only business functions (as in IDEF0). A dynamic model is required, one that accounts for the stochastic nature of request arrivals, service duration, and the limited number of channels (staff).

For the purposes of in-depth analysis, three key business processes were identified as the most representative in terms of assessing branch workload and service quality:

1. Deposit Opening Process-combines digital initiation channels with physical interaction at the branch. Based on source analysis, the process includes the following steps:

– Initiation (Customer)-the customer fills out an online application on the bank’s website.

– Parameter Agreement (Customer)-the application specifies key parameters: term, currency, interest rate, possibility of replenishment and withdrawal.

– Application Processing (Bank)-the application is processed by the system almost in real time.

– Branch Selection (Customer)-the customer chooses the nearest branch to complete the operation.

– Physical Visit (Customer–Bank)-the customer arrives at the branch, where identification takes place.

– Contract Conclusion (Bank)-the bank employee provides the customer with all necessary information, including the information sheet on deposit guarantee, as required by regulatory standards.

For queueing system (QS) modeling, steps 5 and 6 are critical, as they create workload for branch personnel and require service time.

2. Loan Application Processing-this process is more complex, as it is longer and multi-stage, consisting of the following phases:

- Initiation – the customer submits an application (at the branch).
- Verification and Analysis-the bank conducts an assessment of creditworthiness (scoring).
- Stage III (Decision)-either signing the loan agreement and disbursement of funds, or rejection.
- Stage IV (Monitoring)-quarterly monitoring of the borrower's activities.
- Stage V (Repayment)-repayment of the loan and interest.

When analyzing this process, the problem of reconciling time scales becomes evident. The purpose of the model is to analyze "queue and resource management" within the branch during a single operational day. Stages IV (Monitoring) and V (Repayment) last for months or years and do not create a direct workload for the front-office queue (except for repayment operations, which can be classified as "payments").

Therefore, for the purposes of queueing system (QS) simulation, the Loan Application Processing process will be decomposed. The model will include only those parts that consume front-office resources in real time:

- initial consultation of the customer with the loan manager;
- submission and registration of the application;
- signing of the loan agreement (after receiving approval from the back office, which in the model is represented as a time delay).

3. Payment Processing-this is the most common and fastest operation in the branch, consisting of the following steps:

- Initiation (Customer)-the customer submits a payment order or performs the transaction at the cashier's desk.
- Processing (Bank)-the transaction receives the status *Pending*. This is the time the teller/cashier spends entering the data.
- Completion (Bank)-the transaction receives the final status *Success*, and the customer receives a receipt. These three processes (deposit, loan, payment) form a comprehensive set for modeling the workload on different types of resources (tellers and managers) in the banking branch.

The next step, after the qualitative description of processes, is their formalization, i.e., translation into quantitative parameters necessary for building a simulation model that allows for evaluating system efficiency.

Input Parameters-define the external environment and the configuration of the system being modeled. Output Parameters/Key Performance Indicators (KPIs)-metrics calculated as a result of the simulation, allowing the evaluation of system efficiency in accordance with the research objectives.

A generalized formalization of parameters, serving as a "data dictionary" for the simulation model, is presented in Table 2.

For a comprehensive description of the subject area, a textual description and parameter formalization alone are insufficient. It is necessary to apply standardized Computer-Aired Systems Engineering for visual modeling of the system [5].

The use of three different notations-IDEF0, DFD, and UML-is not redundant. They form an "analytical triad," allowing the same system to be described from three orthogonal, complementary perspectives, namely:

– IDEF0 models the functions, control mechanisms, and resources of the system. It answers the question: *"What does the system do, and which rules (Controls) govern it?"*

– DFD (Data Flow Diagram) models data flows and answers the question: *"What information moves, where is it stored, and how is it processed?"*

– UML (Use Case Diagram) models user interaction and answers the question: *"Who (Actors) uses the system and for what purpose (Use Cases)?"*

The IDEF0 notation (Integration Definition for Process Modeling) is used to create a functional model that describes the functions, actions, and processes of an organization or system. The IDEF0 model is hierarchical.

The context diagram A-0 "Conduct Banking Activities" represents the highest level of abstraction (context diagram), showing the system as a single functional block and its interaction with the external environment.

The A0 decomposition diagram represents the next level of the model, detailing block A-0 into key subfunctions that correspond to the identified processes (Fig. 2) and structure (Fig. 1). The arrows between blocks indicate connections (for example, *"Customer Data"* is the output of all three operational blocks and serves as the input for A4, which is responsible for reporting).

Unlike IDEF0, DFD (Data Flow Diagram) visualizes how data (information) moves through the system, rather than which functions are performed. This makes it ideal for modeling the *informational* aspect of our system. The DFD model is also hierarchical.

The DFD context diagram (Level 0), analogous to A-0 in IDEF0, shows the system as a single process and its interaction with external entities (terminators). The detailed DFD (Level 1) diagram decomposes Process 0 into key subprocesses and data stores.

Use Case Diagrams from the Unified Modeling Language (UML) are ideal for modeling the *socio-economic* aspect, as they focus on user goals (actors) and their interaction with the system. The use case diagram of the banking system summarizes what users can do with the system within the branch.

Table 2

Formalization of bank branch model parameters

Category	Parameter	Symbol	Description / Unit	Type
Input (Flows)	Arrival intensity (Deposits)		Number of clients/hr	Stochastic (Input)
Input (Flows)	Arrival intensity (Loans)		Number of clients/hr	Stochastic (Input)
Input (Flows)	Arrival intensity (Payments)		Number of clients/hr	Stochastic (Input)
Input (Resources)	Number of tellers		People	Constant (Controllable)
Input (Resources)	Number of managers		People	Constant (Controllable)
Process Parameters	Service time (Deposit)		Minutes	Stochastic (Input)
Process Parameters	Service time (Loan)	T_{cred}	Minutes	Stochastic (Input)
Process Parameters	Service time (Payment)		Minutes	Stochastic (Input)
Output (KPI)	Average waiting time	W_q	Minutes	Statistical (Output)
Output (KPI)	Average queue length	L_q	People	Statistical (Output)
Output (KPI)	Teller utilization	U_{oper}	%	Statistical (Output)
Output (KPI)	Manager utilization	U_{man}	%	Statistical (Output)
Output (KPI)	Throughput	$N_{processed}$	Applications/day	Statistical (Output)

Source: author's own

Thus, analysis using CASE tools has provided three different yet consistent perspectives on the system, which constitute the necessary foundation for the development of a comprehensive conceptual model.

The conceptual model serves as the central outcome of system analysis, ensuring the transition from a statistical description of the subject area to its dynamic software implementation. Its purpose lies in the generalized reproduction of system behavior over time, the structuring of its main components, and the formalization of the logic of their interaction.

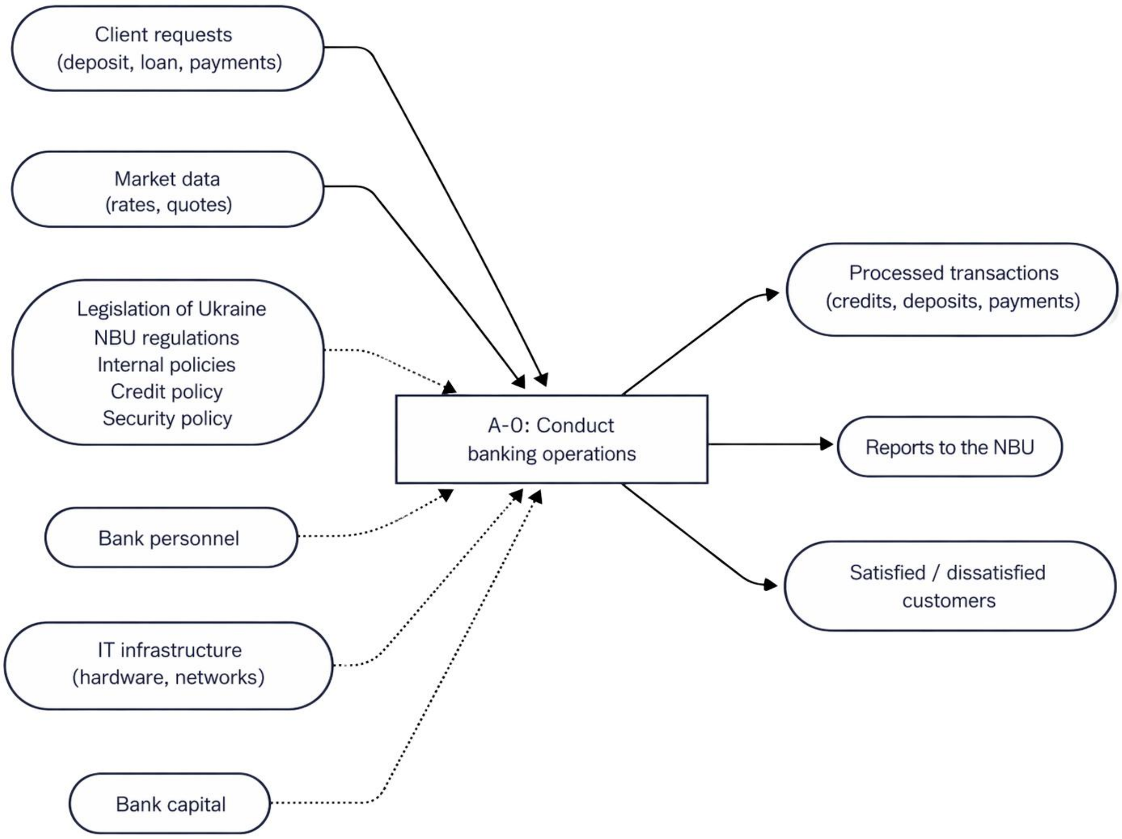

Fig. 1. Context Diagram A-0

Source: author's own

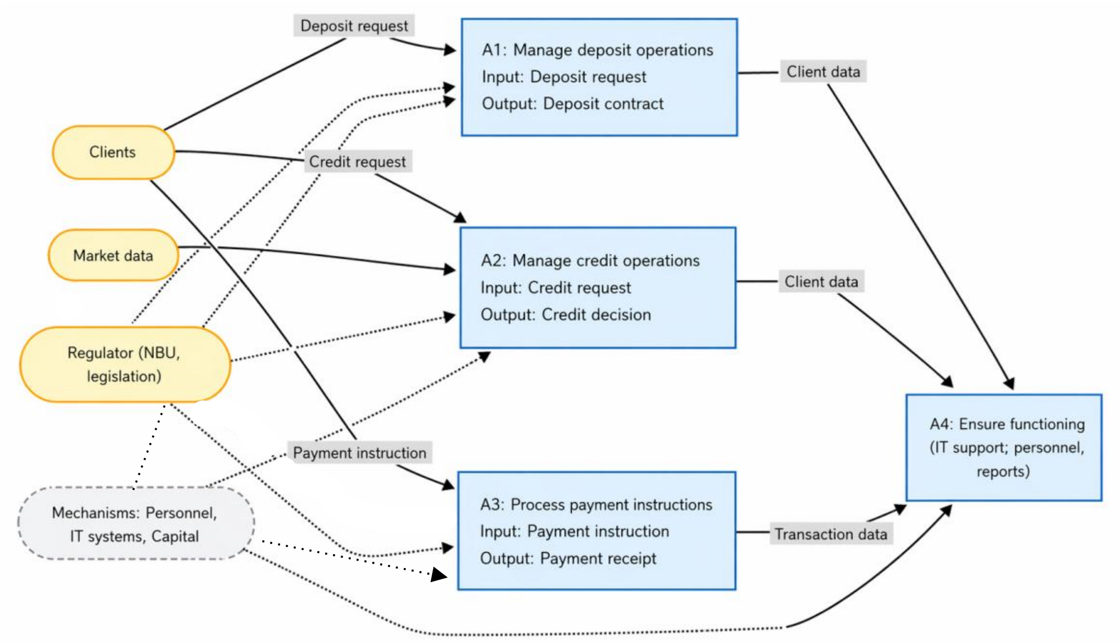

Fig. 2. Decomposition Diagram A0

Source: author's own

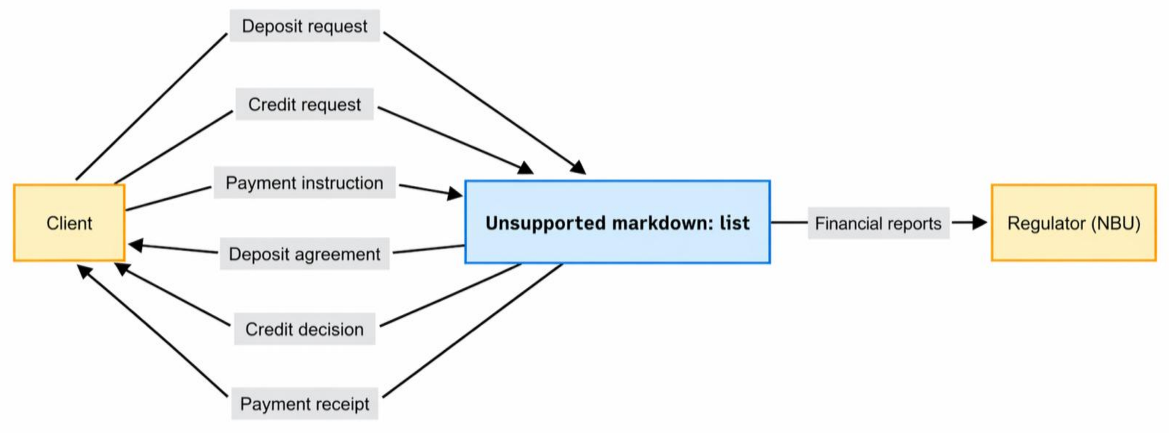

Fig. 3. Context Diagram (DFD Level 0)

Source: author's own

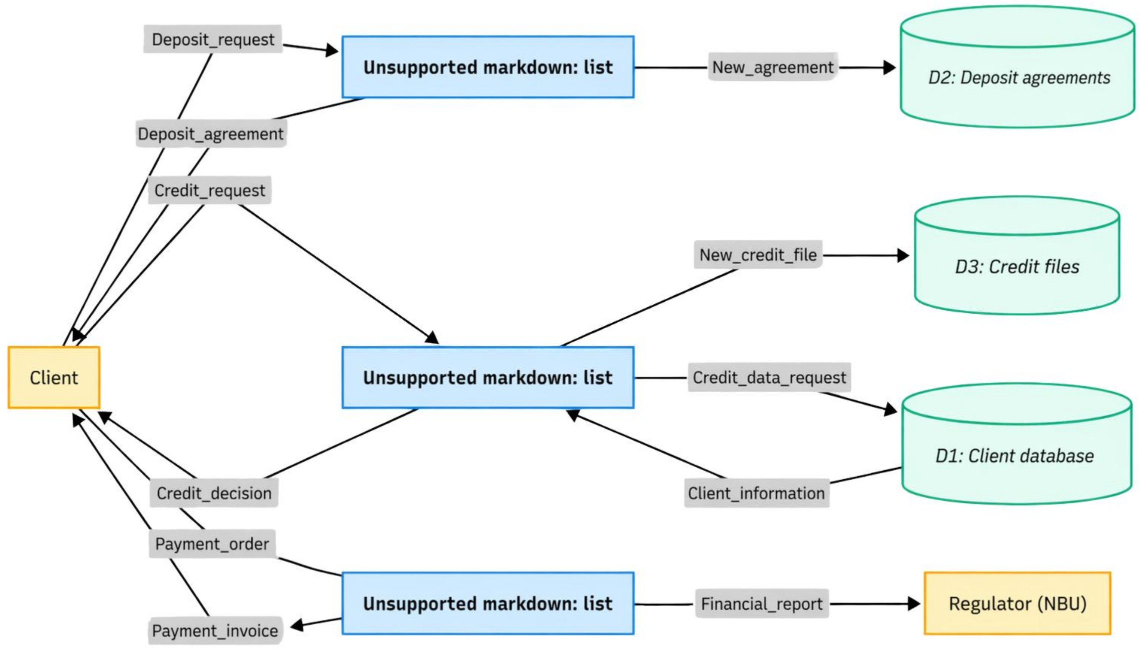

Fig. 4. Detailed Diagram (DFD Level 1)

Source: author's own

Table 3

Explanation of the diagram structure

Element	Description
Actors	Client, Bank Employee, sub-roles – Teller, Credit Manager
System	A rectangle titled "Bank Branch System," which depicts all the use cases
Use Cases	"Open a deposit," "Apply for a loan," "Make a payment," "Verify client identity," "Order credit history"
Include Relationship	"Identity verification" – a mandatory part of opening a deposit and loan processing.
Extend Relationship	"Order credit history" – an optional scenario that can extend the loan application process

Source: author's own

Static diagrams (IDEF0, DFD, UML) show what the system does and what information circulates within it, but they do not answer the question of how the system will behave under load (for example, if 100 customers arrive simultaneously while only one resource is available). The conceptual model transforms structure into behavior by adding key elements to the analysis – time, queues, and stochasticity.

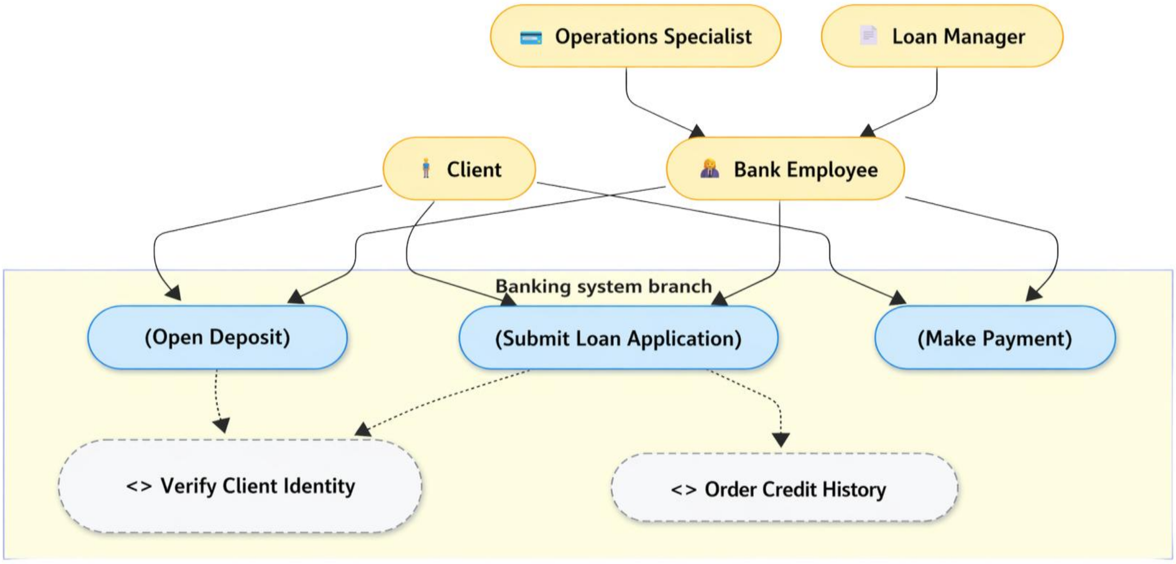

Fig. 5. Use case diagram of the banking system

Source: author's own

Based on the formalization and the identified processes, the conceptual model of the banking branch is described as a multi-channel, multi-phase queueing system (QS) with heterogeneous request flows and resource types.

Table 4

Model boundaries and assumptions

Category	Description
In Scope	Processes in a physical bank branch that require client-staff interaction ("customer service, queue management"). Time horizon – one operational day (or week)
Out of Scope	Macroeconomic factors, long-term credit monitoring, online banking (which does not create queues), back-office operations (only as a source of delays, e.g., $T_{approval}$)

Source: author's own

Table 5

Assumptions

Model Assumptions
Client flows (   ) are independent and follow a Poisson distribution
Service time (   ) follows an exponential or log-normal distribution (to be refined during model calibration)
Queue discipline – FIFO (First-In, First-Out) for each service category
Resources (staff) are interchangeable within their group ($N_{oper} -$ all tellers are equivalent; $N_{man} -$ all managers are equivalent)
Tellers handle deposits and payments; managers handle loans. No cross-functional service is provided (baseline assumption)

Source: author's own

Table 6

Model Entities

Entity	Attributes
Client	ArrivalTime, ServiceType (1=deposit, 2=loan, 3=payment), ServiceTime (individual service time: *Tdep*, *Tcred*, *Tpay*)

Source: author's own

Table 7

Resources

Resource	Description
Tellers (Operators)	Resource pool of size N_{oper}
Managers	Resource pool of size N_{man}

Source: author's own

The described conceptual model constitutes a complete technical specification for the development of a simulation model in any environment that supports the discrete-event paradigm.

Table 8

Operating Logic (Process Flow)

Stage	Description
Generation (Source)	Three separate generators create clients with arrival rates λ_{dep}, λ_{cred}, λ_{pay} and assign them a ServiceType
Routing	If ServiceType = 1 or 3 - Queue_Operators; if ServiceType = 2 - Queue_Managers
Waiting (Queue)	The client waits in the corresponding queue
Seizing (Seize)	Once a resource becomes available, the client seizes one unit of the corresponding resource
Service (Delay)	A delay occurs according to the ServiceTime ($T_{dep}, T_{cred}, T_{pay}$)
Release	The client releases the resource
Exit (Sink)	The client leaves the system; statistics are recorded (total time, waiting time W_q)

Source: author's own

The choice of an appropriate software tool is a critical stage that directly affects the feasibility of implementing the developed conceptual model and achieving the research objectives. This choice must be based on an objective comparative analysis of candidate tools within the context of the specific requirements of the task.

For clarity and justification of the choice, the key candidates are compared (Table 9).

Based on the comparative analysis of simulation tools, it can be stated that the Simulink environment should be excluded from further consideration, as it does not methodologically correspond to the specifics of the task, which is oriented toward modeling queueing systems and the behavioral aspects of subject interactions.

Table 9

Comparative analysis of modeling tools

Generation (Source)	Three separate generators create clients with arrival rates λ_{dep}, λ_{cred}, λ_{pay} and assign them a ServiceType
Routing	If ServiceType = 1 or 3 - Queue_Operators; if ServiceType = 2 - Queue_Managers
Waiting (Queue)	The client waits in the corresponding queue.
Seizing (Seize)	Once a resource becomes available, the client seizes one unit of the corresponding resource
Service (Delay)	A delay occurs according to the ServiceTime (T_{dep}, T_{cred}, T_{pay})
Release	The client releases the resource
Exit (Sink)	The client leaves the system; statistics are recorded (total time, waiting time W_q)

Source: authors' own [1; 6; 7]

At the same time, Arena and Python SimPy can be regarded as effective tools for implementing the basic queueing system model, since they provide adequate reproduction of discrete-event processes underlying the functioning of the studied object.

However, the analysis has established that the most well-grounded choice for conducting this research is the AnyLogic environment. This is due to the fact that the modeled object is considered a complex socio-economic and informational system, for which it is important not only to represent the process (informational) component but also to account for the behavioral characteristics of its elements.

Although the Arena environment demonstrates high efficiency in modeling the process aspect of system functioning, it has limitations in reproducing complex behavioral interactions. In contrast, AnyLogic, through the integration of discrete-event and agent-based approaches, enables deeper and more realistic modeling of socio-economic processes.

In particular, the use of the agent-based approach allows individual agents (for example, customers) to be endowed with specific behavioral characteristics, such as patience level, queue selection strategy, or reaction to changes in service conditions. This, in turn, makes it possible to go beyond classical queuing system models and conduct extended *"what-if"* experiments, which serve as an important tool for supporting critical managerial decision-making.

Conclusion

As a result of the conducted study, it has been established that modern banking systems should be considered as complex socio-economic and informational systems, whose functioning is characterized by a high level of dynamism, stochasticity, and nonlinearity. This determines the limitations of classical analytical approaches and substantiates the appropriateness of using simulation modeling as a key tool of system analysis.

It has been demonstrated that simulation modeling enables the transition from a statistical description of the system to the study of its behavior over time, thereby creating the prerequisites for conducting *"what-if"* experiments and substantiating managerial decisions without interfering with real business processes. It has been determined that, for the study of banking operations, the most relevant paradigm is discrete-event modeling, which allows for adequate reproduction of customer service processes, queue management, and resource allocation. At the same time, the use of the agent-based approach has been found appropriate for accounting for customer behavioral characteristics, while system dynamics is suitable for analyzing the strategic aspects of bank functioning.

It has been substantiated that the digital transformation of the banking sector, in particular the transition to service-oriented and microservice architectures, significantly complicates the architecture of business processes and creates new types of "bottlenecks" associated not only with physical resources but also with digital components (such as service delays and IT infrastructure limitations). This necessitates a comprehensive modeling approach that integrates process, informational, and behavioral aspects.

Within the framework of system analysis, the decomposition of banking activities has been carried out, and the key front-office processes that generate system load have been identified: deposit opening, loan application processing, and payment execution. It has been demonstrated that these processes are representative for evaluating the efficiency of branch operations and can be formalized in the form of a queueing system.

A methodological approach to building the simulation model has been formulated, which includes the identification of input flows, the description of queue and service mechanisms, and the definition of a system of key performance indicators. A formalization of model parameters has been proposed, ensuring the creation of a consistent *"data dictionary"* for subsequent software implementation.

It has been shown that the use of standardized Computer-Aired Systems Engineering (IDEF0, DFD, UML) makes it possible to form a multidimensional representation of the system, encompassing the functional, informational, and behavioral aspects of its operation. On this basis, a conceptual model of the banking branch has been developed as a multi-channel, multi-phase queuing system with heterogeneous request flows and resources, accounting for temporal dynamics, queues, and the stochastic nature of processes.

The choice of the modeling environment has been substantiated. It has been established that specialized discrete-event simulation tools (Arena and Python SimPy) are suitable for reproducing the process logic of the system, but they have limitations in modeling complex agent behavior. It has been demonstrated that AnyLogic is the most appropriate tool, as it supports hybrid modeling by combining discrete-event, agent-based, and system-dynamics approaches, thereby providing a more comprehensive and realistic representation of the system under study.

Thus, the formulated conceptual and methodological framework creates the foundation for the further development of a software simulation model of the banking branch, the execution of computational experiments, and the derivation of quantitatively substantiated recommendations for optimizing operational activities and improving customer service quality.

References:

1. Tomashevskyi, V. M. (2005). *System Modeling: Textbook*. Kyiv: BHV Publishing Group, 352 p. (in Ukrainian)
2. Volkova, N., & Popyk, A. (2022). Analysis of financial stability management of JSC CB "PrivatBank." *Economy and Society*, no. 45. (in Ukrainian)
3. Derevyanko, N. A., & Kostkin, K. K. (2025). Transformation of banking lending practices in Ukraine under conditions of high uncertainty: Analysis of trends 2022–2024. *Current Issues of Economic Sciences*, no. 13. Available at: (in Ukrainian)
4. Danylyuk, I. (2026). Information-analytical and financial security foundations of strategic management of the credit market under conditions of economic transformation. In *International Security Studies: Managerial, Technical, Legal, Environmental, Informational, Economic and Psychological Aspects*. Lublin: ISAP, Research and Education, vol. I, pp. 93–128
5. Danylyuk, I. V., & Duma, L. V. (2021). Features of applying CASE technologies. In *Fifty-eighth Economic and Legal Discussions:* Proceedings of the International Scientific and Practical Online Conference (Lviv, June 24, 2021). Lviv, pp. 11–12. (in Ukrainian)

6. Rockwell Automation. (2024). Arena Simulation Software. Available at: Available at: https://www.arenasimulation.com (accessed April 24, 2026)
7. MathWorks. (2024). Simulink Documentation. Available at: https://www.mathworks.com/help/simulink/ (accessed April 24, 2026)