BSDS: Balance Sheet Dynamics Simulator

Volker Liermann and Harro Dittmar

1 Introduction

The idea of agent-based modeling has a long history. Actor models and graph-theory-oriented algorithms that propagate complex systems of many interacting entities through time have existed for more than half a century (Hewitt 1969). Along with the concept of cellular automata, they emerged as tools for the rather esoteric field of complexity science (Conway 2000). However, this bottom-up approach stayed dormant for a while until it gained popularity in economics. This can be attributed to recent developments in distributed computation, the performance of modern computational infrastructures, the availability of large open-source programming libraries and the growing number of established model references. It is safe to say that ABMs are still the new kid on the block in the model zoo of the financial sector risk manager. Their design as actor models is suitable for a scalable bottom-up approach oriented toward the micro-states and micro-interactions of the participants in a financial system. By design, such models are very flexible regarding successive refinement of the agent behavior.

V. Liermann (✉) · H. Dittmar
ifb SE, Grünwald, Germany
e-mail: Volker.Liermann@ifb-group.com

H. Dittmar
e-mail: Harro.Dittmar@ifb-group.com

V. Liermann and C. Stegmann (eds.), *The Digital Journey of Banking and Insurance, Volume II*, https://doi.org/10.1007/978-3-030-78829-2_8

The goal of the Balance Sheet Dynamics Simulator (BSDS), the core ideas of which have been described in (Liermann and Dittmar 2021), is to simulate the balance sheet dynamics of banks in a competitive macroeconomic environment. This chapter focuses on technical details of the implementation of the BSDS as an actor-oriented model that uses the Simudyne software development kit[1] and a Hadoop environment for distributed computation.

1.1 Financial Navigator and Other Applications

The BSDS addresses the new transaction development aspects of a stress test. For this purpose, it generates scenarios of the balance sheet development that complement the analysis of risks associated with changes in the balance sheet structure. The BSDS is integrated into the Financial Navigator (see Thiele 2021), a scenario-based credit risk management framework aiming to discover the mid-term impact (3–5 years) of rating changes driven by macroeconomic scenarios on a projected portfolio[2] that has possible applications in planning and controlling.

Recent conceptual improvements in the planning process aim to make the process more dynamic to accelerate the response to changing environments. We refer to this trend as value-driver-oriented planning.[3] Value-driver-oriented planning involves the identification of drivers behind the planning volumes and making them (or more holistically the generated values for the institute) dependent on observable indicators. The first step on the road to a value-driver-oriented planning process is to establish a connection (maybe by using regression models) between the driver and the targeted transaction volume in the balance sheet. This approach is always backward looking and cannot incorporate new trends or behavioral changes. The BSDS predicts possible structural changes. From the results, impulses could be derived to adjust the mechanics connecting the value driver and the projected volumes in the balance sheet.

The BSDS is meant as a prototype for a challenger model for the estimates made by the planning process (conventional or value-driver-oriented). Another application could be asset liability management in banks, especially

[1] For details see Simudyne (n.d.).

[2] Actual portfolio plus the simulated new transaction generated by the BSDS.

[3] See Valjanow et al., Digital Planning—Driver-Based Planning Levaraged by Predictive Analytics (2019) and Valjanow et al. (2021).

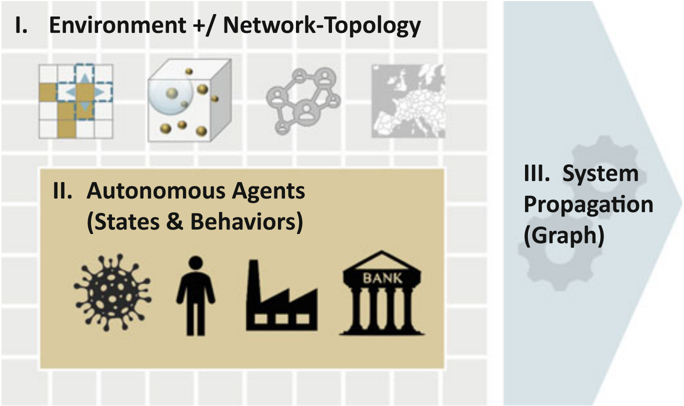


Fig. 1 ABM—essential elements (© ifb SE)

the aspect of the interest rate in the banking book[4] The BSDS presented here is a model state with a reduced functional scope for didactic purposes.

1.2 Structure of the Chapter

This chapter is structured as follows: Sect. 2 describes the basic assumptions and model components, Sect. 3 introduces the technical details of Simudyne's ABM framework and the Hadoop[5] cluster on which the simulation runs are performed, Sect. 4 discusses some preliminary results regarding the calibration of parameters and Sect. 5 gives an outlook for sensible extensions of the model.

2 Model

2.1 Recap: Elements of an Agent-Based Model

Figure 1 shows the essential ingredients that are needed to define an ABM for a specific use case. Those are the agent classes, the environmental definitions and the action sequence that defines the rules for the propagation of the system through time.

[4] The regulatory requirements are given by (Basel Committee on Banking Supervision, Interest rate risk in the banking book 2016) and this standard is integrated in 2021 Basel Framework (Basel Committee on Banking Supervision, The Basel Framework 2021).

[5] Hadoop is a specialized computing cluster designed to store and analyze large amounts of unstructured data (for more details see Akhgarnush et al. 2019).

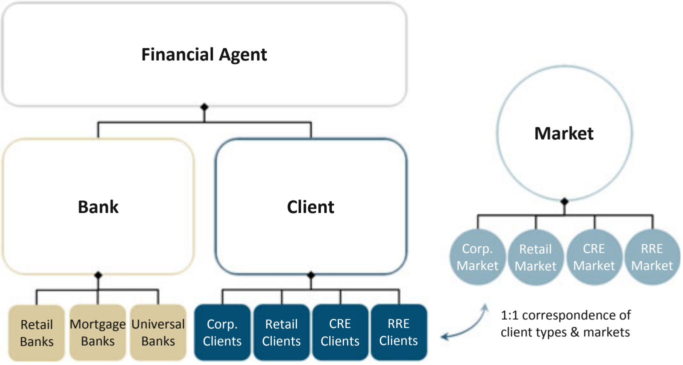


Fig. 2 BSDS—hierarchy of agents in a class diagram (© ifb SE)

Agents are defined as autonomous computational entities that interact with their environment (including other agents). From the technical viewpoint, they are instances of classes with (1) field variables, the values of which represent an agent's state and (2) algorithms that encode their possible actions (behavior). Agents interact by exchanging messages and the behavioral code reacts to those messages by updating the state of the agent. Effectively, agents thereby change states of themselves and induce state changes of other agents.[6]

Obviously, the structure that defines the possible agent interaction as exchange of messages is another essential ingredient of an ABM. It is referred to as topology of the computing graph that specifies the links over which the agents send the information to interact.

While the agent topology defines which agents can interact, the action and messaging sequence defines the order of computational tasks and communication itself. It coordinates the agents' behavior and propagates the system from one time step to the next, with synchronization of the agents' activity status at the end of each time step (see Pregel approach to graph processing [Malewicz et al. 2010]). The following subsections explain these three essential ingredients of an ABM for the specific case of the BSDS.

2.2 Agent Types

The BSDS has three major agent types: banks, clients and markets. Figure 2 shows the corresponding hierarchy of agent classes.

Banks and clients inherit common properties (but no action code) from the financial agent class, such as variables that store their type, ID and a list

[6] See mode details in Sect. 3 in (Liermann and Dittmar 2021).

of loan agreements. The bank and client classes differ considerably in their actions: while banks try to grow and establish a certain portfolio structure, clients merely seek a certain total amount of credit depending on individual desired debt ratios and the global macroeconomic situation.

The different subtypes of banks and clients exhibit the same behavior among their subclass, but their algorithms use different parameters: for example, retail banks have a higher contingent for retail clients in their portfolio, and corporate clients apply for loans with longer durations than retail clients.

For each of the four client types, there is one market, reflecting the simplifying assumption that clients apply for only one client-type-specific product and do not seek diversified portfolios. The markets form non-discretionary matches between the individual supplies and demands of banks and clients by combinatorial optimization. The only difference between the markets are their links to other agents, as explained in the following section.

2.3 Environmental Definitions

The environmental definitions of the BSDS do not include any spatial aspects. The topological environment of an agent is defined by the links to the agents it communicates with, as shown in Fig. 3.

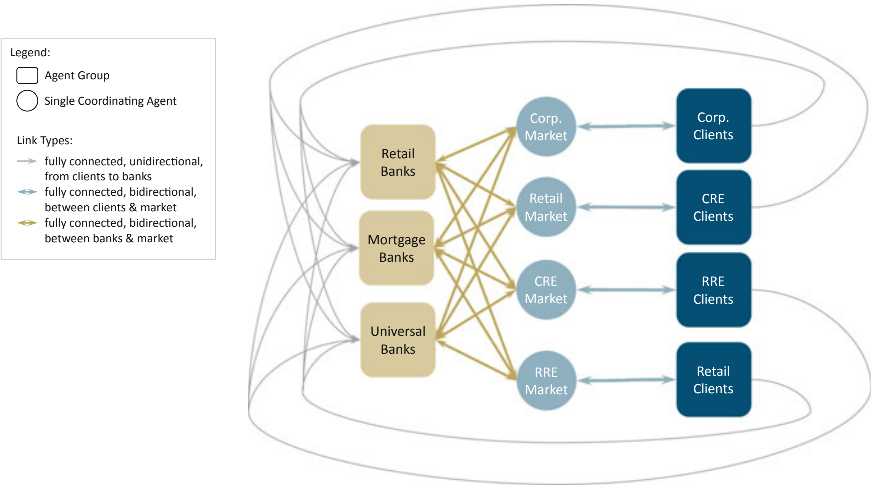

Legend:
Agent Group
Single Coordinating Agent
Link Types:
fully connected, unidirectional, from clients to banks
fully connected, bidirectional, between clients & market
fully connected, bidirectional, between banks & market
Retail Banks
Mortgage Banks
Universal Banks
Corp. Market
Retail Market
CRE Market
RRE Market
Corp. Clients
CRE Clients
RRE Clients
Retail Clients

Fig. 3 BSDS—topology of directional communication links between groups of agents (© ifb SE)

The groups of different agent classes are fully connected, reflecting the simplifying assumption that no bank or client has an advantage or disadvantage during the fair and rational matching of supplies and demands that is controlled entirely by the markets. This simple topology corresponds to a world with unrestricted sharing/homogeneous availability of information. Clients and banks both interact with markets to communicate their individual credit supplies and demands, and to receive a response whether a suitable business partner has been found during the matching process (blue and golden arrows in Fig. 3). Since no rating agencies are modeled, clients communicate their current rating information directly to the banks (grey arrows).

Another environmental element is the time series of an exogenous simulation of the macroeconomy, meaning they follow their own, decoupled dynamics that are independent of the agent topology. Macroeconomic parameters can be accessed globally but remain unaffected by the agents. Individual supplies and demands only depend on the current agreements and the exogenous macroeconomy.

The number of agents and their connectivity are kept constant throughout the simulation. At this stage, we do not consider any long-term growth or network dynamic aspects of the system but assume these to be negligible on the time scale of our simulations.

Note that the market agents play a special practical role in this topology: depending on the number of bank client agents and the logic behind the market interactions, the number of messages that must be sent between actors to cover all interactions possible in principle in a fully connected network can be very large. Therefore, it is necessary to reduce this number by introducing "coordinating" agents. Having more than one market increases the efficiency of the overall action sequence by reducing the number of messages that need to be sent between the financial agents: instead of having to process individual communication between all possible $(N_{banks} \cdot N_{clients})$ pairs of banks and clients, markets cache $(N_{banks} + N_{clients})$ messages and process them in one action. In this role, the markets are referred to as "coordinating agents", as they funnel and reduce the flux of information through the system. This improves the performance, if the tasks that need to be done for each pair are not so complex or heterogeneous that they would actually benefit from being processed by financial agents in parallel threads instead of being processed sequentially by a centralized market.

2.4 Sequence of Agent Actions and Messages

The action sequence that propagates the system through time is shown in Fig. 4. It defines the course of actions that are carried out at each time step with a temporal resolution of 1 month. The sequence is repeated iteratively to generate the evolution of bank portfolios over a time span of 36 to 60 months. The colors indicate which agent groups execute the actions described. For example, dark blue action blocks are performed by all members of the four client groups in parallel. Some of the action blocks contain several numbered actions, which are performed one after the other (without needing to be triggered individually). For example, the market begins to match individual supplies and demands after having collected all loan applications and exposure reports. Action blocks are arranged as split blocks or sequence blocks. The actions of a split block start simultaneously, while the actions of a sequence block start successively from left to right.

Each time step starts with client and market actions: clients update their desired debt ratios according to the GDP growth factor and apply for loans that are calculated from this debt ratio, the running loans and their respective client size. Simultaneously, clients send their current ratings to the banks, which then update their desired total exposure and use the latest client ratings to compare their target rating structures with the actual ones according to their current portfolios and their bank-type-specific business models. The resulting exposure gaps translate to the credit supplies that are reported to the four markets. The markets match credit demands and supplies with a parameterized combinatorial optimization algorithm and send response messages

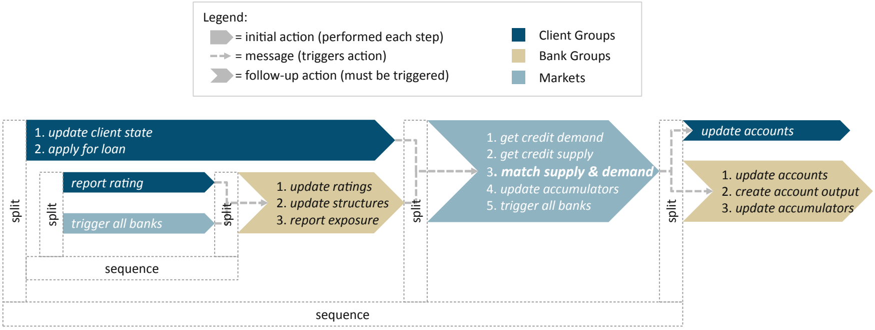
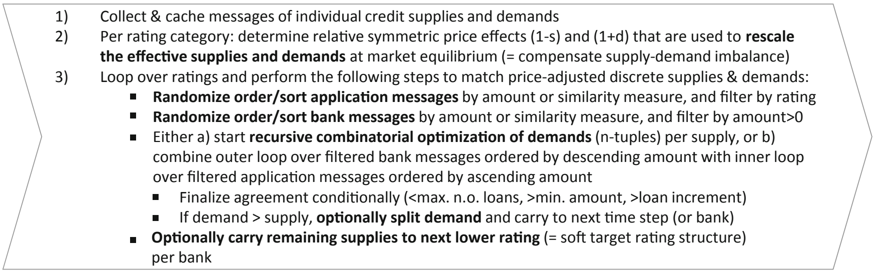


Fig. 4 BSDS—sequence of actions and messages of the actor model. The diagram shows blocks of actions that are executed by all individual members of the corresponding color-coded agent group (© ifb SE)



1) Collect & cache messages of individual credit supplies and demands

2) Per rating category: determine relative symmetric price effects (1-s) and (1+d) that are used to **rescale the effective supplies and demands** at market equilibrium (= compensate supply-demand imbalance)

3) Loop over ratings and perform the following steps to match price-adjusted discrete supplies & demands:
- **Randomize order/sort application messages** by amount or similarity measure, and filter by rating
- **Randomize order/sort bank messages** by amount or similarity measure, and filter by amount>0
- Either a) start **recursive combinatorial optimization of demands** (n-tuples) per supply, or b) combine outer loop over filtered bank messages ordered by descending amount with inner loop over filtered application messages ordered by ascending amount
    - Finalize agreement conditionally (<max. n.o. loans, >min. amount, >loan increment)
    - If demand > supply, **optionally split demand** and carry to next time step (or bank)
- **Optionally carry remaining supplies to next lower rating** (= soft target rating structure) per bank

Fig. 5 BSDS—outline of the steps performed by the matching algorithm of the markets (© ifb SE)

to the respective counterparties of successful, non-discretionary matches. The logic behind this matching routine (action "3. match supply & demand" in Fig. 4) is outlined in Fig. 5, and parameterized behavior is marked bold.

Note that credit default events are not considered in this model, and cashflows are not modeled as explicit independent action events, but only represented implicitly in pre-defined maturity structures of the exposures. Instead of generating loans in a parametric form that would specify payment schedules, they are generated as complete exposure time series with a client-type-specific static structure. Accordingly, it is instructive to think of the matching mechanism for discrete incongruent amounts as an "exposure Tetris".

2.5 Measures of the Matching Efficiency

Three time-averaged indicators are proposed as measures of the effectiveness of the matching algorithm, or the "performance of the exposure Tetris":

1. E: the total **exposure of banks** to clients in active agreements,
2. L: the average number of actual **loan agreements** per client, which must stay below the pre-defined upper bound,
3. M: the market-averaged ratio of successfully **matched loan amounts** relative to the sum over all requested amounts, ranging from 0 to 100% for complete matching and
4. I: the bank-type-averaged **incongruity of target and actual rating structures**, ranging from 0 for perfect overlap to 1 for complete incongruity.

Using the indices r for rating class, c for client, m for market (equivalent to client type), b for bank and t for bank type, the following formulas can be used to describe the respective contributions per time step:

$$L = \frac{1}{N_{clients}} \cdot \sum_{c} l_c$$

Equation 1: Average number of actual loan agreements

$$M = \frac{\sum_{c_m} \left(f_{c_m} \cdot a_{c_m} \right)}{\sum_{c_m} a_{c_m}}$$

Equation 2: Ratio matched loan amounts

$$I = \frac{1}{3} \cdot \sum_{t} \left(\frac{1}{N_{banks,t}} \cdot \sum_{b_t} \left(\sum_{m} \sum_{r} \left| f_{b_t,m,r} - g_{b_t,m,r} \right| \right) \right)$$

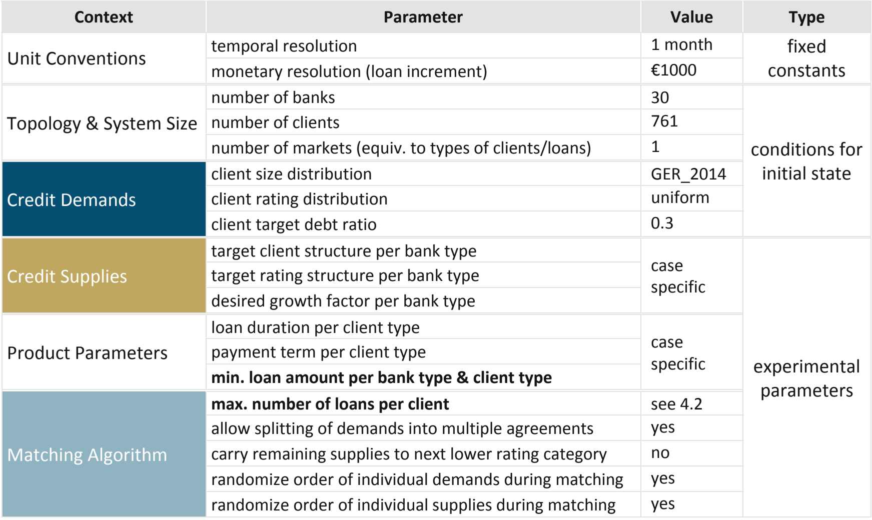
Equation 3: Averaged incongruity of target and actual rating structures

l_c is the number of loans of client c. a_{c_m} are the individual requested loan amounts of the clients of type m. The multiplicators f_{c_m} that range from 0 to 1 represent the success of individual loan applications considering the possibility that only a fraction of the requested amount was matched as a consequence of splitting it into agreements with multiple banks. $f_{b_t,m,r}$ are the actual fractions of the banks' total exposure in one client type and rating category, and $g_{b_t,m,r}$ are the target fractions according to the bank-type-specific business model.

2.6 Limiting Conditions and Model Parameters

In general, adding complexity to a model complicates the interpretation of simulation results. The more parameters are involved, the less universal the model and the larger the calibration or balancing effort. Table 1 lists essential parameters of the model, grouped by six contexts and categorized either as fixed constants, conditions for the initial state or experimental parameters. The fixed constants correspond to fundamental model assumptions. The initial conditions are parameters that specify limiting conditions for the generation of the artificial initial state (see Sect. 4.1). The experimental parameters can be adjusted in successive experimental simulation that runs to tune the system behavior such that it exhibits non-trivial behavior and mimics trends of real systems.

Table 1 BSDS model parameters (© ifb SE)



Context	Parameter	Value	Type
Unit Conventions	temporal resolution	1 month	fixed constants
	monetary resolution (loan increment)	€1000	
Topology & System Size	number of banks	30	conditions for initial state
	number of clients	761	
	number of markets (equiv. to types of clients/loans)	1	
Credit Demands	client size distribution	GER_2014	
	client rating distribution	uniform	
	client target debt ratio	0.3	
Credit Supplies	target client structure per bank type	case specific	experimental parameters
	target rating structure per bank type		
	desired growth factor per bank type		
Product Parameters	loan duration per client type	case specific	
	payment term per client type		
	min. loan amount per bank type & client type		
Matching Algorithm	**max. number of loans per client**	see 4.2	
	allow splitting of demands into multiple agreements	yes	
	carry remaining supplies to next lower rating category	no	
	randomize order of individual demands during matching	yes	
	randomize order of individual supplies during matching	yes	

The abbreviation “GER_2014” refers to the income tax distribution according to the German wage and income tax statistics from 2014, which was taken from the Statistisches Bundesamt (Lohn- und Einkommensteuer 2014, Statistisches Bundesamt [Destatis], Artikelnummer: 2140710147005 2018).

3 Technical Implementation

3.1 Technical Scenario

Figure 6 summarizes the core elements of the technical scenario for running the simulations. All mandatory platform and software requirements are shown along with the numbers of compatible versions.

Simudyne offers a software development kit that facilitates the setup of agent-based models and provides a convenient browser-based graphical user interface for the control of simulation runs. Alternatively, simulation runs of a model that has been launched on Simudyne’s “Nexus Server” can be controlled via calls to a REST-API. To enable the model to access input data for the initialization of the system from SAP Hana tables, we imported the JDBC source driver recommended by SAP, and to write parquet output to Hive tables, we imported the Apache Hadoop library. The model was

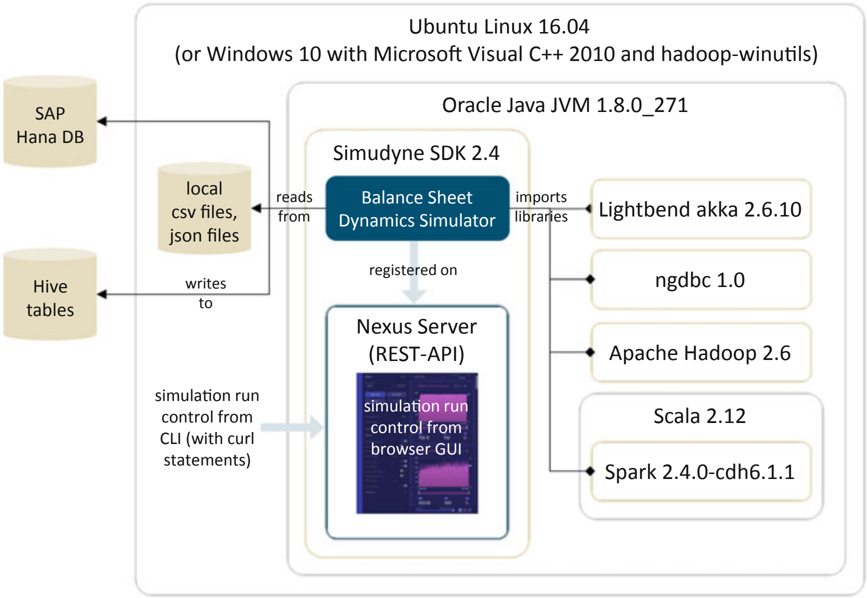

Ubuntu Linux 16.04
(or Windows 10 with Microsoft Visual C++ 2010 and hadoop-winutils)
Oracle Java JVM 1.8.0_271
Simudyne SDK 2.4
SAP Hana DB
local csv files, json files
Hive tables
reads from
writes to
Balance Sheet Dynamics Simulator
registered on
Nexus Server (REST-API)
simulation run control from browser GUI
simulation run control from CLI (with curl statements)
imports libraries
Lightbend akka 2.6.10
ngdbc 1.0
Apache Hadoop 2.6
Scala 2.12
Spark 2.4.0-cdh6.1.1

Fig. 6 BSDS—details of the technical implementation and code dependencies (© ifb SE)

programmed in Oracle Java 8 with IntelliJ IDEA that used Maven 3.6.3 for compilation.

3.2 Simudyne

The BSDS was programmed as an actor model using the Simudyne software development kit (SDK) version 2.4, which adopts the Pregel approach to graph processing (Malewicz et al. 2010) and uses Spark as a cluster management tool and Akka to coordinate the actor-based parallel computing (communication between agents). Akka relieves the user from creating the infrastructure for the actor system with the low-level code necessary to control basic behavior that is independent of the use case, such as the synchronization of messages (for further information see Lightbend [n.d.]). A detailed documentation of the Simudyne SDK is available on the website of the vendor (Simudyne n.d.). The supported Spark version depends on Scala 2.12.

3.3 Hadoop Cloudera (Distributed Calculation)

The compiled model was tested in the local Windows 10 environment, which required the installation of Microsoft Visual C++ and hadoop-winutils, and on a Hadoop cluster that runs Ubuntu Linux 16.04, is maintained with the Cloudera manager, and had access to four virtual machines that were rented on the Microsoft Azure cloud. For this article, we only tested the parallel execution of multiple instances of the model itself, which makes the sampling of multiple possible system trajectories scalable. In our continued collaboration with Simudyne we plan to distribute the model graph of the BSDS itself over several machines in the Azure cloud. This will also make the system size scalable so that it will be possible to increase the number of agents to tens of thousands and more.

4 Ergodicity and Preliminary Results

This section discusses first results for one central aspect of the calibration of parameters. When developing a bottom-up model from scratch with parameters and degrees of freedom that are based on pure intuition, it is necessary to balance their influence on the model and check whether they all add value to the model in terms of required complexity and realism. Here, we focus on the calibration of two parameters of the matching algorithm with the goal to increase the effectiveness of the matching algorithm and maintain the ergodic behavior of the system. These two parameters are the influence of the minimum offered loan amount and the maximum allowed number of loans per client. One can also think of an economic interpretation of these parameters: the lower bounds for the loan amounts represent the fact that a loan only becomes attractive once its return is significantly larger than the associated costs. Similarly, the limitation of the number of loans per client can be interpreted as the tolerance of clients toward the burden associated with the administration of multiple accounts.

4.1 Initial States

Figure 7 gives an overview of the input data that is required for the model initialization and the output data that is generated during the simulation run.

For consistency with the assumptions that we have made to simplify the portfolio, it is necessary to generate synthetic loan data that enables the initialization of the model in a corresponding simplified state. For systems

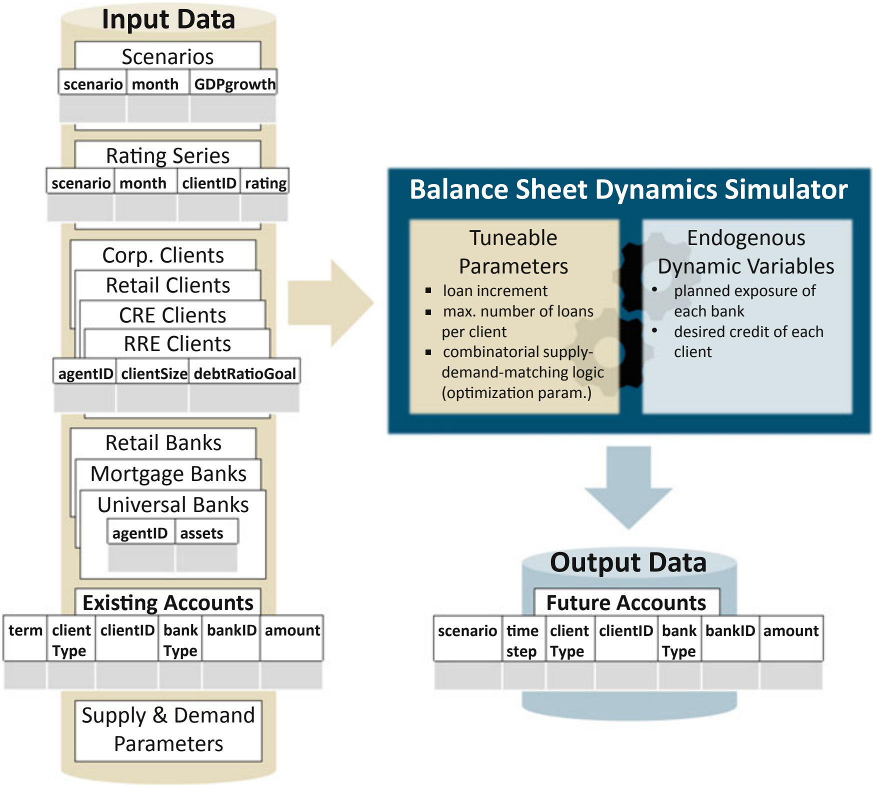


Fig. 7 BSDS—Input for system initialization and output of simulation runs (© ifb SE)

that exhibit a sensitive dependence on initial conditions, picking only one arbitrary artificial initial state might not lead to valid statistical results,[7] which is why it is important to repeat simulation runs starting from an ensemble of independent initial conditions. This gives indications for ergodic system behavior that allows a comparison of time averages instead of ensemble averages (Poitras und Heaney 2015). To compare steady states, one must meet the conditions for mixing that result in ergodic system behavior. This condition for meaningful time-averaged steady-state values also applies to the performance indicators that we use to analyze the matching efficiency.

The initial distribution of remaining terms reflects certain assumptions about the dissipation of the exposure sum over time, which should be generally reproducible by the simulation. This means that an artificial initial state that did not emerge from the model itself must be given the chance to relax to

[7] Since the BSDS is still in a developmental stage, we skipped the rigorous analysis of the aspect of ergodicity and merely ensured that the development of the loans “looked chaotic enough”.

a steady state in a pre-run with fixed macroeconomy, client ratings and target exposures (see Sect. 4.2). Using such states as a reference, one can attempt systematic studies of stylistic macroeconomic influences on the system.

To test the matching algorithm, the model complexity was reduced by disabling all but the corporate clients. Simulation runs start from a hypothetical initial state with a total of €5.7 trillion in exposures of 30 banks that are distributed among the accounts of 761 clients in such a way that the target rating structures are perfectly matched at the start of the simulation. Accordingly, the initial values of the indicators are $L = 1$ and $I = 0$. The distribution of amounts for the artificial loan data approximates the income distribution according to the German wage and income tax statistics from 2014, which was taken from the Statistisches Bundesamt (Lohn- und Einkommensteuer 2014, Statistisches Bundesamt [Destatis], Artikelnummer: 2140710147005 2018).[8] The creditworthiness (expressed in a credit rating) of the client was taken from a uniform distribution due to the lack of real client data. Further, it was assumed that all target debt ratios lie above the initial debt ratios by 30% to start from a state of excess demand and a busy matching routine.

4.2 Simulation Runs

During the simulation, the exposure sum drops as loans mature and are not fully replaced with new ones. Several factors cause these maturing exposures to give rise to incongruent demands and supplies. Their interplay enables the matching process to introduce the **chaotic mixing of the credit amounts** that are required for aperiodic system behavior and the associated quasi-continuous shifts of exposure distributions:

1. **Asymmetric agent behavior**: There are different conditions for the messaging of loan applications and exposure reports to the markets. Thus, the sum over supplies that have been restored does not equal the sum over demands of clients who reapply with some lag.
2. **Possibility of mismatches** or sub-optimal matches of the discrete amounts per time step: The association of loan amounts with bank–client pairs changes during the simulation. Also, there is the possibility to split demands and match them with supplies from multiple banks.

[8] The loan amount distribution, annual return distribution and private income distribution are different things, but the reasonable assumption that their shapes have similar features suffices for a proof of concept.

3. **Discretization of loan amounts**: There is a minimum loan increment and a minimum amount below which applications are not accepted.
4. **Iterative approximation of target exposure structures**: Restored supplies are redistributed over client type and rating categories to approximate/stay near the exposure structure of the business model.

The price adjustment that is applied to compensate for the difference between total demand and total supply in each time step introduces further deviation from the initial equality of total supplies and demands. Considering all these cutting and scaling elements of one iteration, chaotic mixing seems like a reasonable assumption, underpinned by the evolution of the total exposure in the system as shown in Fig. 8.

Due to the remainder of mismatched supplies and demands, the value of E drops below its initial value and fluctuates around steady-state levels that reflect the matching efficiency. Table 2 shows how all four proposed indicators of the matching efficiency change when

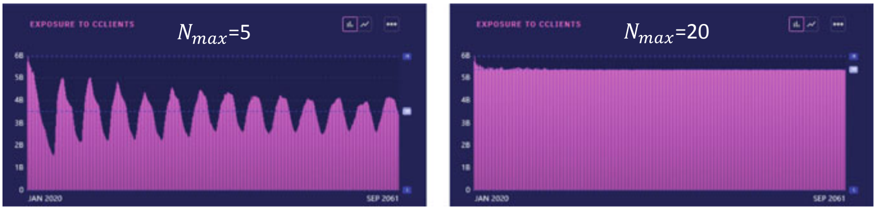


Fig. 8 BSDS—evolution of the total exposure in the system in a pre-run over 500 iterations. On the left side, the number of loans per client is limited to a maximum of 5, on the right side to a maximum of 20 (© ifb SE)

Table 2 BSDS results for the efficiency of the matching algorithm (gold columns) at different combinations of parameters (blue columns). The indicators were averaged in the steady states that were reached after 250 iterations (© ifb SE)

N_{max}	A_{min} (€M)	E(€T)	L	M	I
5	1	4.8	5.3	100%	0.11
10	1	5.0	9.7	100%	0.10
15	1	5.4	14.0	100%	0.06
20	1	5.4	17.0	100%	0.06
50	1	5.7	17.7	80%	0.04
20	50000	5.6	14.1	80%	0.06
20	500000	5.0	4.9	50%	0.12
20	5000000	2.6	0.3	10%	0.94

A. the number of loans at which clients stop applying, N_{max}, is increased from 5 to 20 and
B. the minimum accepted loan amount, A_{min}, is increased from 1 thousand to 5 billion euros.[9]

For small N_{max}, there were not enough successful matches per time step, which prevented efficient mixing and left periodic patterns in the exposure structures that would not dissipate on a reasonable time scale (limited system size artifacts). Consequently, no steady state was established that would allow the exploration of comparably subtle effects of macroeconomic scenarios. However, allowing clients to satisfy their demands with a larger number of individual agreements eliminated this effect and the system reached a steady state within less than 250 iterations, in which the total exposure did not drop much below its initial €5.7T.[10]

The increase of A_{min} shows that a good approximation of the rating structure of the business model was still possible with clients (groups) that apply for loan amounts of at least €50000M. That means, on the stylized scales and under the assumptions of our model, it would not have an adverse effect on the portfolio planning of the banks if they were to ignore loan requests below €50000M. Increasing the minimum loan amount above this value had the undesirable effect that it excluded a significant portion of applications from the matching. Consequently, M decreased, I increased and L dropped.

When the matching did not use random combinations of demands and supplies, but performed a combinatorial optimization of successive demand triples that leave the smallest gap when filling up the supplies, a higher total exposure of €5.69T could already be reached at $N_{max} = 20$. However, this also introduced an ordering effect that caused a periodic evolution of M and a significant increase of the averaged rating incongruity I to 0.21. That means if the market uses this mechanism to yield the best coverage of the desired exposure, more banks need to be flexible to deviate from their target rating structure.

[9] Note that the clients' demands were scaled up to account for the unnaturally large ratio of the number of bank agents to the number of client agents that was used to boost the performance of these preliminary tests. To compensate for this imbalance, clients apply for very high loan amounts ranging from about €5M to €5B.

[10] The fraction of matched demands per time step, M, decreases to some extent, because the fluctuation of the individual amounts of supply and demand becomes larger relative to the amounts that have not been matched in previous steps.

5 Summary and Outlook

5.1 Summary

In this chapter, we presented a prototype for an agent-based model that simulates how the balance sheets of banks adapt to changes in the environment, as represented by macroeconomic parameters that are linked to client ratings. These environmental changes act as boundary conditions of a matching algorithm that re-allocates credit supplies among a constant set of clients by performing a simple “exposure Tetris”. A combinatorial optimization algorithm matches individual credit supplies and demands under the influence of macroeconomic effects, causing the rating structures of the predicted portfolios to adapt.

Our results show how important it is to tune parameters in the context of simplifying assumptions. For example, for the high ratio of bank to client numbers that results from limiting the total number of agents, one also needs to allow more loans per client, N_{max}, to enter a regime in which the system behaves more naturally. A sensible economic interpretation is that we effectively consider client groups and not individual clients. The parameters of the matching algorithm define its efficiency, which could be successfully monitored by measures like the achieved total exposure in the system and the average incongruence of actual and target rating structures of the bank portfolios. Due to the small size of the system of under 800 agents, changing the parameters N_{max} and A_{min} also had a significant influence on mixing properties and thus on the fundamental system behavior. It is therefore a necessary step to make the system large enough and find regimes in which a limiting behavior with smoother amount distributions is established. This will make it possible to study more subtle aspects such as disruptive changes of client ratings.

5.2 Reasonable Model Extensions

With our future efforts we wish to reproduce real trends and aid the identification and optimization of systemic (model-level) parameters that can be used to find the best compromise between reaching the target exposure and maintaining the target rating structure in different macroeconomic scenarios. As the calibration and balancing of model parameters progresses, we hope to reproduce trends that can be expected in real bank portfolios. This will allow us to explore how different macroeconomic scenarios affect the deviation of the actual rating structures of bank portfolios from their business model, and which business models are the most resilient or competitive ones in different scenarios. Thanks to the flexibility of the ABM design, the adaptation of ideas for the elaboration of the model is straightforward to implement into the code without compromising the existing elements. Some of our planned investigations are:

- Reduce limited size effects by increasing the number of clients to ensure strong chaotic mixing that facilitates the evaluation of steady-state properties,
- Generate artificial client and account data with more realistic features regarding the distribution of ratings and loan amounts among the clients per client type,
- Define rules for partially connected network topologies with additional markets that limit the bank–client pairs that are available for matching: this will enable a better compromise between fully informed coordinating agents and high parallelization of matching processes and
- Incorporate rating models that are coupled to the system evolution and enable a dynamic client set (spawning of new competitors and deleting of defaulted client agents).

Of course, one must always consider whether such extensions of the complexity also require additional real data and complicate the calibration of parameters, and whether additional simplifying assumptions are compatible with those that have already been made.

Literature

Akhgarnush, Eljar, Lars Broeckers, and Thorsten Jakoby. 2019. "Hadoop—A Standard Framework for Computer Clusters." In *The Impact of Digital Transformation and Fintech on the Finance Professional*, edited by Volker Liermann and Claus Stegmann. New York: Palgrave Macmillan.

Basel Committee on Banking Supervision. 2016. *Interest Rate Risk in the Banking Book.* Basel: Basel Committee on Banking Supervision.

———. 2021. January 22. Accessed January 30, 2021. https://www.bis.org/basel_framework/.

Conway, John. 2000. *On Numbers and Games.* New York: CRC Press.

Hewitt, Carl. 1969. "PLANNER: A Language for Manipulating Models and Proving Theorems in a Robot." *IJCAI'69: Proceedings of the 1st International Joint Conference on Artificial Intelligence*, May: 295–301.

Liermann, Volker, and Harro Dittmar. 2021. “BSDS—Balance Sheet Dynamic Simulator (Application ABM).” In *The Digital Journey of Banking and Insurance, Volume I—Disruption and DNA*, edited by Volker Liermann and Claus Stegmann. New York: Palgrave Macmillan.

Lightbend. n.d. *Getting Started Guide.* Accessed December 15, 2020. https://doc.akka.io/docs/akka/2.6/guide/.

———. 2018. *Lohn- und Einkommensteuer 2014, Statistisches Bundesamt (Destatis), Artikelnummer: 2140710147005.* Statistisches Bundesamt.

Malewicz, Grzegorz, Matthew Austern, Aart Bik, James Dehnert, Ilan Horn, Naty Leiser, and Grzegorz Czajkowski. 2010. *Pregel: A System for Large-Scale Graph Processing.* Google.

Poitras, Geoffrey, and John Heaney. 2015. “Classical Ergodicity and Modern Portfolio Theory.” *Chinese Journal of Mathematics*, February 6: 1–17.

Simudyme. n.d. *Simudyme.* Accessed December 15, 2020. https://portal.simudyne.com.

———. n.d. *Welcome to the Age of Simulation.* Accessed December 15, 2020. https://www.simudyne.com/.

Thiele, Markus. 2021. “Financial Navigatorc—A Modern Approach to Analytical Banking.” In *The Digital Journey of Banking and Insurance, Volume I—Disruption and DNA*, edited by Volker Liermann and Claus Stegmann. New York: Palgrave Macmillan.

Valjanow, Simon, Philipp Enzinger, Daniel Suttner, and Maik Alexander Schmidt. 2021. “Value-Driver-Oriented Planning—Management-Oriented Design and Value Driver Identification.” In *The Digital Journey of Banking and Insurance, Volume I—Disruption and DNA*, edited by Volker Liermann and Claus Stegmann. New York: Palgrave Macmillan.

Valjanow, Simon, Philipp Enzinger, and Florian Dinges. 2019. “Digital Planning—Driver-Based Planning Levaraged by Predictive Analytics.” In *The Impact of Digital Transformation and Fintech on the Finance Professional*, edited by Volker Liermann and Claus Stegmann. New York: Palgrave Macmillan.