Graph Models for Capital Markets

Nneka Chinelo Ene¹, Maribel Fernández¹, and Bruno Pinaud²

¹ King’s College London, UK
Nneka.Ene@kcl.ac.uk Maribel.Fernandez@kcl.ac.uk
² University of Bordeaux, France
Bruno.Pinaud@u-bordeaux.fr

Abstract. The financial crisis of 2008 has heightened the need for more effective and transparent tools in the modelling of capital markets. In this paper we propose to use strategic port-graph rewriting (a particular kind of graph transformation system, where port-graph rewriting rules are controlled by user-defined strategies) as a visual modelling tool to analyse credit derivative markets. We illustrate the methodology by specifying and analysing a “rational negligence” model, whereby investors may choose to trade securities without performing independent evaluations of the underlying assets. The model has been implemented within PORGY — an interactive port-graph rewriting tool that is able to assist with the analysis of simulation results.

1 Introduction

The sub-prime mortgage crisis of 2008 has heightened the need for more effective and transparent tools in the modelling of uncertainty of returns from expected cash flow streams within the capital markets. Improper evaluation of new mortgage derivative products is believed to have been a key driver behind the bubble. Given the failure of traditional top-down macro equilibrium models to predict the crisis, economists are gradually turning away from them in favour of more autonomous agent-based models (ABM), that, as a result of examining behaviour at a micro-level, are able to provide more realistic views [6].

Rational negligence has been noted as the behavioural pattern among traders in the Asset-Backed Securitisation (ABS) space that has most led to weakening in the market, whereby the desire to not exercise proper due diligence in favour of operational costs (supported by strong liquidity in the market despite negative underlying macroeconomic circumstances), was greatly subscribed to. Ratings from credit agencies were found to be inaccurate especially in terms of underestimated default probabilities, based on improper assessments of the underlying [1], details the DSGE (Dynamic Stochastic General Equilibrium) models were unable to capture or anticipate.

In this paper we seek to formalise this key theory using a graph transformation approach. We identify a series of rules and strategies that guide the operational execution of a graph transformation system hierarchically representing the ABS universe, right from origination to placement of securitised assets within
a secondary market. By employing a graph transformation paradigm, in-built mechanisms that favour accurate verification and validation, or assurance, of ABM, can be utilised. In addition, not only is there a marked underdevelopment of modelling tools that handle interrelatedness within entities in the market [12], the advantage presented by being able to visually trace the evolution of our system right up to equilibrium is especially useful given the well-known challenges in complex system modelling.

We present a hierarchical model that captures the key details of the rational negligence phenomenon [1, 9], whereby investors may choose to trade securities without performing independent evaluations of the underlying assets. The top tier deals with asset transfers, and beneath the top tier implemented lie several operational tiers that at a basic level encapsulate asset origination, packaging and structuring details. The model has been implemented in PORGY\(^3\), an interactive, port graph rewriting tool. A verified and validated base case provides an effective platform for incrementally increasing the complexity and scope of the model, and for performing key experiments to identify parameters that could mitigate or trigger a crisis. In the long-term, we shall seek to investigate the propagation speeds of negative and positive sentiment in capital markets, under varying parameters, especially regarding the sensitivity of underlying asset toxicity or likelihood of default, and develop a framework for monitoring sellers pay-offs. Such a framework can also have cross-industry applicability especially in other areas of banking where collections of contractual and metrical rules contribute greatly to work-flows and processes.

**Related Work.** In addition to the above mentioned agent-based market model [1], other operational models have been developed such as the multi-agent model by Markose et al. [13] that focuses on credit risk transfer in Residential Mortgage Backed Securities (RMBS) with Credit Default Swaps (CDS). The Multi-Agent Financial Network (MAFN) model [11] is able to effectively monitor bank activity and stress test policy for perverse incentives. Despite considerable efforts, holistic visualisation and monitoring of the financial system in order to identify systemic risks however remain a visible need within Agent-Based Economics.

Graph Transformation Systems have been used as a basis for the development of modelling tools in many areas: for example, SESAM (Software Engineering Simulation by Animated Models) [15] is an application of graph grammars to the analysis of the software engineering process; RuleBENDER\(^4\) is a simulation tool that supports rule-based modelling and is compatible with biological ODE and PDE (ordinary and partial differential equation) projects written in the BioNetGen Language [16]; PORGY [2, 17] has been used to model social networks and biochemical processes.

This paper is organised as follows: We shall firstly examine key notions useful in our analyses in Section 2. The general approach that shall be taken with regards to the modelling of securitisation shall be addressed in Section 3. Rules

\(^{3}\) [http://porgy.labri.fr](http://porgy.labri.fr)

\(^{4}\) [http://www.rulebender.org](http://www.rulebender.org)
representing our chosen model and associated strategies are given in Section 4. Section 5 examines key properties of the model and we finally conclude by summarising key takeaways and briefly outlining future plans in Section 6.

2 Background

2.1 Asset-Backed Securities

As defined in [12] “Securitisation is the process of converting cash flows arising from underlying assets or debts/receivables (typically illiquid such as corporate loans, mortgages, car loans and credit cards receivables) due to the originator into a smoothed liquid marketable repayment stream” and this ensures that the originator can raise asset-backed finance through loans or the issuance of debt securities.

An originator is any financial intermediary with a portfolio of assets on its balance sheet. Assets represent loans to clients or obligors who make regular installment payments to the originator to clear their debts. In a securitisation, assets are selected, pooled and transferred to a tax neutral, liquidation-efficient (i.e. bankruptcy avoiding), special purpose entity (SPE) or special purpose vehicle (SPV), who funds them by issuing securities. Securitisation issues backed by mortgages are called MBS (mortgage-backed securities) where the mortgages can either be residential or commercial, securitisation issues backed by debt obligations are called CDOs (collateralised debt obligations). In general, an ABS (asset-backed security), or simply asset if there is no ambiguity, is any securitisation issue backed by consumer loans, car loans, credit cards, etc.

Securities are rated by credit agencies (e.g., Standard and Poor’s) that employ varying methodologies in their analyses. The securities are bought by investors (usually banks or other financial agents), who are paid back over time by the SPV from the stream of receivables. Investors can then trade these ABSs, characterised by varying levels of liquidity based on the type of asset, with others in the secondary market. Figure 1 summarises the securitisation process.

Aside from identification attributes, Table 1 defines the attributes associated with the major entities relevant to our model’s development.

With regards to the core rational negligence model [1], the profit $U_w$ expected by an agent (e.g., a bank) $w$ from trading an asset depends on whether or not $w$ follows the negligence rule, i.e., the rule of not performing independent risk assessment. Let $z$ be a binary variable indicating whether or not the agent is following the negligence rule, then $U_w(z)$ can be characterised by the following equations, where $p$ is the probability of asset toxicity, $Z$ is the mean of all $z$’s in the domain, $c$ is the cost of purchasing an asset (the payoff from successfully reselling the asset is normalised to unity), $x_w$ is the cost of performing a complete due diligence analysis.

- Expected profit for $w$ when following the rule, i.e., when $z(w) = 1$, if $w$ buys an asset and then tries to sell it to $w'$:

$$U_w(1) = -p(1 - z(w'))c + [1 - p(1 - z(w'))](1 - c) = 1 - p(1 - Z)c$$
This is because if the asset is toxic then \( w \) will lose \( c \) if \( w' \) checks, and will have a profit of \( 1 - c \) if \( w' \) does not check. Of course \( w \) does not know a priori whether \( w' \) will or not follow the rule, but it can estimate \( z(w') \) as the average of all the values of \( z \) in the system, \( Z \).

− Similarly, the expected profit for \( w \) when the rule is not followed, i.e., \( z(w) = 0 \), is defined by:

\[
U_w(0) = (1 - p)(1 - c) - x_w
\]

This is because if the asset is toxic, then \( w \) will not buy it (loosing only \( x_w \)), but if it is not toxic then it will resell it with a profit of \( 1 - c = x_w \).
So the best response of agent \( w \) to a buying request is determined by:

\[
U(1) - U(0) = p(Z - c) + x_w = p \left( \frac{1}{k} \sum_{j \in N_i} z_j - c \right) + x_w
\]

The goal is to study the evolution of the system till fixed point or stable state is reached i.e., in this case, a state such that all potential buyers in the universe of discourse no longer alternate between diligent and non-diligent behaviour in their handling of the purchase of a particular asset.

2.2 Graph Transformation Systems

There are many different kinds of graph transformation systems see, for instance, [4,5,10,14]. In this paper we shall examine the transformation of port graphs [7], which have been used as a modelling tool in various domains (see, e.g., [16,17]).

Port Graphs. Intuitively a port graph is a graph where nodes have explicit connection points, called ports, and edges are attached to ports. Nodes, ports and edges are labelled by a set of attributes. Port graphs are transformed by applying port graph rewriting rules.

A port graph rewriting rule \( L \Rightarrow C R \) can itself be seen as a port graph consisting of two port graphs \( L \) and \( R \) together with an “arrow” node. The pattern, \( L \), is used to identify subgraphs in a given graph which should be replaced by an instance of the right-hand side, \( R \), provided the condition \( C \) holds. The arrow node (or more precisely, the ports and edges that connect the arrow node to the left and right-hand sides of the rule) describes the way the new subgraph should be linked to the remaining part of the graph, to avoid dangling edges during rewriting. When the correspondence between ports in the left-hand and right-hand sides of the rule is graphically obvious, we may omit drawing the ports and edges involving the arrow node.

For example, the port graph in Figure 2 depicts a typical ABS secondary market universe represented by a community of banks (B), some of which own tradeable assets (A), the rating agencies (R), and the originators (O) connected to the asset. Transactions between banks, representing buy-sell requests and associated operations can then be specified by means of rules. Figure 3 shows one of the rules used in our model, implemented in PORGY (see section 4). In this paper, we will study the behaviour produced by the trading of one asset; the number of assets can be increased to perform real data validations but one asset is sufficient to perform validations against equivalent DSGE analyses.

Port Graph Rewriting. Let \( X \) and \( Y \) be two port graphs over the same signature \( \nabla \). A port graph morphism \( f: X \rightarrow Y \) maps nodes, ports and edges of \( X \) to those of \( Y \) such that the attachment of ports and the edge connections are preserved, and all attributes are preserved except for variables in \( X \), which must be instantiated in \( Y \). Intuitively, the morphism identifies a subgraph of \( Y \) that
is equal to \( X \) except at positions where \( X \) has variables (at those positions \( Y \) could have any value). We refer to [7,8] for formal definitions and examples.

We denote by \( f(X) \) the subgraph of \( Y \) consisting of the set of nodes, ports and edges that are images of nodes, ports and edges in \( X \). This definition ensures that each corresponding pair of nodes, ports and edges between \( X \) and \( Y \) have the same set of attribute labels and associated values, except at positions where there are variables. When using this definition to define rewriting, we will only allow the use of variable labels on one of the graphs: \( X \) will be the graph on the left-hand side of the rewrite rule, which may include variable labels, and \( Y \) will be the graph to be rewritten, without variables.

Let \( G \) be a port graph. A rewrite step \( G \Rightarrow H \) via the port graph rewrite rule \( L \Rightarrow_C R \) is obtained by replacing in \( G \) a subgraph \( g(L) \) by \( g(R) \), where \( g \) is a morphism from \( L \) to \( G \) satisfying \( C \). More precisely, we say a match \( g(L) \) of the left-hand side (i.e., a redex) is found if: there is a port graph morphism \( g \) from \( L \) to \( G \) (hence \( g(L) \) is a subgraph of \( G \)), \( C \) holds, and for each port in \( L \) that is not connected to the arrow node, its corresponding port in \( g(L) \) must not be an extremity in the set of edges of \( G - g(L) \). This last point ensures that ports in \( L \) that are not connected to the arrow node are mapped to ports in \( g(L) \) that have no edges connecting them with ports outside the redex, thus ensuring that there will be no dangling edges when \( g(L) \) is replaced by \( g(R) \). A sequence of rewriting steps is called a derivation.

**Strategic Rewriting.** Strategies in rewriting systems are means of controlling the creation of rewriting steps. Given an initial graph, a strategy allows us to select the rewrite steps that are relevant to our model. For a given graph, several dif-
A derivation tree is a collection of derivations with a common root.

In PORGY, the strategy language allows us to specify not only the rule to be used in a rewriting step, but also the position where the rule should (or should not) be applied. A located graph consists of a port graph with two distinguished subgraphs: a position subgraph and a banned subgraph, denoted $G^P_Q$. Rewrite rules can only be applied to $G$ if they match a subgraph which superposes $P$ and does not superpose $Q$. Thus, the position and banned subgraph guide the application of rules by indicating where rewriting can and cannot take place, leading to concise traversal algorithms. A table summarising the syntax of PORGY’s strategy language can be found in the Appendix.

PORGY [7], our implementation system, is a port-graph rewrite engine that offers an in-built strategy editor, a navigable derivation tree widget, widgets for the creation of rules and graphs and various visual and non-visual algorithms related to the two classes of objects (see Fig. 3 for a snapshot). Simulations can be created, executed and monitored in various ways including statistically. It supports probabilistic rewriting, and has been used successfully in the implementation of biochemical calculi, interaction nets and social networks.

3 The ABS-GTS Model

In this section we give a high-level description of a graph-based model of the ABS process as specified by the equations given in Section 2.1. We can represent the full ABS universe hierarchically as several initial graphs and enforce information
flow bidirectionally. Port graph rewriting rules and strategies shall be used to control the step-wise evolution of the graphs and to create a derivation tree that can be used for plotting and analysing parameters. The asset trading model sits at the top level of the model hierarchy. It is non-deterministic in nature, though fixed points exist (where the system remains stable), depending on the behaviour the market has adopted with regards to independent analysis given due diligence costs, asset value and liquidity perceptions [1]. Below this system, able also to handle asset pricing and valuation issues, lie several more deterministic subsystems that model origination, structuring, SPV transfers and profitability of the sale, and therefore aid in enforcing internal checks as a result of complete system integration.

Figure 4 summarises all the major components of a securitisation as currently recorded in the system. B is an agent, the buyer or seller bank, A the asset, O, another agent, distinguished by red ports, can represent, in the context of this information flow: the originator, servicer or SPV. Pool represents the loans selected as the underlying for the asset and key analytics such as weighted average prepayment or default rates or seasoning can be computed and stored here, deal structure related details such as tranching and payment distribution are contained in the node called Structure, and Pricing encapsulates all pricing details. AL, AP and AB represent associated loan, property and borrower information respectively. The leaf nodes such as the nodes corresponding to individual loans for example have not been included in this diagram.

Figure 4. All Tiers Flattened and Condensed

Secondary Market Tier. Several processes characterise the secondary market tier (the model’s top level tier); in our system these are represented by means of port graph rewriting rules. The first set of rules handles the core asset transfer
process as a series of rewrite steps, namely: a buy request from the seller to a buyer, an analysis step where the buyer assesses whether or not an independent risk assessment before purchase will be profitable, and execution steps based on the internal analysis. The second set of rules handles rating agency processes that update the pe or perception variable by either increasing/upgrading it or decreasing/downgrading it.

Other Tiers. Operational tiers handle loans, borrowers, SPV and deal/asset structuring. These tiers are a lot more operational and deterministic; suitable loan cherry picking and stratification within various pools of varying vintage, loan application acceptances and servicer repossession action once a pre-outlined number of borrower defaults have occurred are some of the processes that characterise this phase.

In the rest of the paper we focus on the top tier level, which is where the ‘rational negligence’ phenomenon leading to the 2008 crisis can be observed.

4 The ABS-GTS Rational Negligence Model in PORGY

Model execution begins with a parameterised initialisation phase that produces a sample universe with one asset, linked to the owner bank (see Figure 2). Colour attributes in nodes and ports are used to distinguish between classes of objects and to aid in the identification of states of interest, such as negligent behaviour.

Details of the evolution of the system are captured by an internal derivation tree which can be viewed or traced as it evolves by use of a small multiples animation functionality available in PORGY. Fixed point is identified using a Change node, with a check after each trading cycle, as explained below.

Rewrite Rules. Table 2 describes the main rewrite rules handling asset transfer in our model; due to space restrictions the core rules are shown in the Appendix. As in the foundational paper [1], our current implementation has been limited to the investigation of the trading of one asset among k banks. In future this number can be increased in keeping with real data. Many of the rules in Table 2 can be broken down into even more finer grained local steps for increased conceptual validation support; however, this level of abstraction is sufficient to implement the equational model and observe the rational negligence phenomenon.

Execution Strategy. Reduction strategies define sub-graphs to be selected for evaluation and which rules should be applied. The starting state of the model consists of all the agents noted in the ABS background section: the banks (B), originator/servicer (O), the asset (A) and the rating agencies (R) as seen in Figure 2. It is from this point that the derivation tree begins to undergo construction as the execution strategy calls on rules that create step-wise transformations. After the initialisation phase, the asset transfer processes are governed by the strategies AllTrade (see Strategy 1.1) and FixedPointSearch (see Strategy 1.2).

A basic description of the strategy AllTrade is as follows: Line 1 sees to the positioning of the starting state or graph. Line 2 starts a trading cycle: each
iteration of the “repeat” body corresponds to one transaction (the number of iterations is bound by the number of banks, $k$). Line 3 begins an asset transfer operation, with the rules described in Table 2: the owner of the asset offers the asset to another bank; the strategy operator \textit{one} that controls the application of the rule \textit{requesttobuy} ensures non-deterministic choice of buyer; the potential buyer then begins the analysis in line 4 to decide whether or not to follow the negligence rule. It does this by computing the profitability of choices using the parameters probability of toxicity, due diligence cost and current value of the asset and the attribute $Z$ or average sentiment indicator, as described in Section 2.1. The rule used in Line 5 updates the global $Z$ as described in Table 2. The attribute \textit{theta} indicates if it is more profitable to follow or not follow the negligence rule, based on a difference in projected value of the two proposals: if $\textit{theta} \geq 0$ the deviation rules will apply, otherwise the bank follows the negligence rule (see the “orelse” in lines 6 and 7). Note that the \textit{one} operator finds one instance of the left-hand side of the rule within the current position graph in which matching is permitted, as opposed to the \textit{all} operator.

Strategy 1.1 controls the full execution: \textit{AllTrade} is iterated until there are no changes in the agent behaviours (i.e., as long as the \textit{change} rule can be applied).
Strategy 1.1: FixedPointSearch
1 AllTrade;
2 while(match(change))do(
3 one(change);
4 AllTrade
5 )

Strategy 1.2: AllTrade
1 setPos(crtGraph);
2 repeat
3 (one(requesttobuy);
4 one(beginanalysis);
5 one(updatez);
6 (one(deviationresult);one(deviationdecision)) orelse
7 (one(followresult);one(followdecision))
8 )(k)

5 Model Properties

Following a successful base case validation in which test results line up with results from a traditional ABM simulation as noted in [1], we examine parameters and parameter values that simulate different economic scenarios. A natural question arises: What events could have mitigated or further instigated the 2008 crisis? By increasing toxicity values for example we can take into account the increase in interest rates that led to increased default rates and the 2008 crisis. We observe that in this case, the system reaches a stable state where all banks perform independent risk analysis. Apart from high toxicity parameters, what else could enforce a non-rule following equilibrium? By analysing the evolution of the system under different parameter values we seek to uncover other attributes in addition to toxicity that contribute to the extent the rule is followed. The following properties of the implementation ensure the validity of our model wrt the equational model. More precisely, we now show that our implementation is correct and complete with respect to the given equational semantics, that is, the equilibrium states computed by our model correspond to those defined by the equations underlying the rational negligence model of securitisation, as presented in [1]. This ensures that our model correctly captures the ABS process of interest, and predictions from the ABS models under the same conditions coincide with the predictions produced by our system.

Theorem 1. Strategies 1.1 (FixedPointSearch) and 1.2 (AllTrade) never fail.

Proof. Strategy 1.2 (AllTrade) sets the position for rewriting (this command cannot fail), and executes a command of the form repeat(S)(k), which according to PORGY’s semantics [7, 8] iterates the strategy S while it succeeds and
always returns id (i.e., it can never fail). The number $k$ indicates the maximum number of iterations of the loop (in our case, this corresponds to the number of banks). Since AllTrade cannot fail, strategy FixedPointSearch can only fail if the rule change in the body of the while loop fails, which is impossible due to the condition in the “while” (there is at least one match for change). □

The following property ensures that if our model reaches a stable state, then the ABS equational model (see Section 2.1 and [1]) reaches the same stable state.

**Theorem 2 (Correctness).** The graph-based model defined by the initial state, rewrite rules and strategies defined above is correct with respect to the equationally defined ABS process in [1]. More precisely, the graphs generated by the application of the rewrite rules with the given strategy correspond to states reached by the system governed by the equational ABS model.

**Proof.** Let $w$ be the bank that owns the asset (i.e., the bank linked by an edge to the asset), and let $w'$ be the randomly chosen potential buyer (selected by the application of the rule requesttobuy). Rule beginanalysis computes the value of the projected profitability made by $w'$ following and not following the negligence rule using the attributes p-tox, c-val and dd-cost (i.e., probability of toxicity, current value and due diligence cost) in the asset, which correspond to the values of $c$, $x$ and $p$ in the equational model. The result of this computation is the values specified by the equations:

$$U_w(1) = -p(1 - z(w'))c + [1 - p(1 - z(w'))](1 - c)$$

and

$$U_w(0) = (1 - p)(1 - c) - x_w$$

The best response, given by $U(1) - U(0)$, is accurately computed in the attribute theta of node Theta (the indicator value of best choice). The strategy ensures that the potential buyer selects the most profitable choice (lines 6-7 of Strategy 1.2), and the rule updatez recomputes the global Z value, as follows:

$$Z_i + 1 = Z_i * (k - 1) + z(w')$$

which is equivalent to the value specified in the equational model:

$$Z = \left( \frac{1}{k^i} \right) \sum_{j \in N_i} z(w)$$

where $k_i$ is the cardinality of the set of selling agents and $N_i$ the number of agents in the universe. The case in which $N_i$ and $k_i$ coincide (represented in our model by the value $k$) is a centralised system.\footnote{The case of a non-centralised universe can also be accurately modelled in PORGY, using the propagation models developed for social networks [17].}
Hence, each iteration of the repeat in Strategy 1.2 correctly computes the parameters associated to an asset transfer according to the target model. It then follows that the final result of our chosen system is correct with respect to the target model.

**Theorem 3 (Completeness).** The graph-based model defined by the initial state, rewrite rules and strategies defined above is complete with respect to stability as specified by the equational ABS model in [1] (see Section 2.1). More precisely, if the equational model reaches a stable state, so does our model.

**Proof.** The transactions of the equational ABS model are mimicked by the iterations of the “repeat” in our strategy. A stable state in the equational model indicates that banks do not change their approach to negligence, which corresponds to absence of “Change” in our model.

**Theorem 4.** The graph program consisting of the initial graph, rewrite rules and strategy described above terminates if and only if stable state is reached.

**Proof.** A state is stable if no bank has changed its mind regarding its negligence choice when given an opportunity to trade. If stable state has been reached, there is no change after executing AllTrade hence the while loop found in line 2 of Strategy 1.1 stops. Conversely if our strategy terminates then the change rule does not apply since this is the condition to exit the while, hence no bank has changed its behaviour in AllTrade.

In particular, for high values of $p$ (that is, high probability of toxicity), we observe the expected result when the initial state contains a mixture of negligent and diligent agents: a sharp drop in $Z$, corresponding to a sharp switch in average approach (i.e., more banks decide to perform independent analysis), which in turn will generate stability. However, even for high toxicity, if the initial state is a set of negligent agents, the model reaches equilibrium without switching approach.

**Theorem 5 (Negligent equilibrium).** If in the initial graph $Z \approx 1$, then the system arrives at negligent equilibrium with respect to the agents’ choice of whether or not to perform independent risk analysis (i.e., stable state where all banks follow the negligence rule) even with high toxicity.

**Proof.** For high $p$ the profitability equation outlined in section 2.1 reduces to:

$$U(1) - U(0) \approx Z - c + x_w$$

given that the difference between $U(1)$ and expected profit when the rule is not followed (i.e. $U(0)$) is $p(Z - c) + x_w$. This linear equation will be computed at each iteration of the repeat loop. The result must be positive given that $c$ and $x_w$ are both positive constants smaller than 1.

Similarly, we can see that if $p$ is high but in the initial graph the majority of banks are deviating from the negligence rule, then the system reaches a due
diligence stable state. We can also infer from the model conditions to avoid a market crash, such as the one below (that although not feasible in a real market, is valid in equational models).

**Theorem 6 (Indefinite propagation).** A continuous increase in the number of negligent bank agents will mean that a market crash can be postponed.

*Proof.* The creation of arbitrary nodes at run-time can be easily implemented by our chosen tool in addition to the creation of a specified number of clones. The proof that this can aid in the creation of a non-terminating “perfect” system is as follows: A continuous increase in the number of agents used in calculating the average current sentiment, $Z$, as computed by the “updatez” rule above and as outlined in section 2.1 means that the value of $Z$ used in deciding whether or not to perform an independent risk analysis can potentially remain unchanged for selected bank agent parameters regardless of propagation. □

Other useful analyses involve calculating propagation speeds (i.e., number of steps it takes for rule sentiment to be adopted by all agents relative to the size of network) and results of the aggregation of conflicting asset performance signals or the cases of expected falls in cost which can also be modelled by our system.

6 Conclusion

Compared with the previous agent-based models of the ABS process, the main difference is that whilst ABMs rely on the internal processing of its agents, global matching in GRSs is state-based and at each point in time provides a holistic view of the environment. At any point in time it is easy to trace which actions are responsible for which outcomes, a useful feature in models in which even a partial verification poses an enormous challenge.

We hope to make future efforts towards the integration of a more comprehensive and hierarchical system able to more operationally capture all details of the full securitisation life-cycle and the building of internal verification processes. Fixed point at the top-level to include details of pay-offs and gains within each participatory institution and lower levels designed to address various loan cherry-picking securitisation rules, repline initialisation rules, borrower initialisation rules and rules and strategies for general analytics, should be able to cater for this leading to a greater possibility of full system integration and built-in verifications. As noted by John Bachman and Peter Sorger “For combinatorially complex systems, equation-based models are hard to error-check, extend and reuse, in contrast to rule-based models, which are concise, comprehensible and easily extended. Research to date suggests that rule-based approaches enable simulation and analysis of classes of complex reactions that would otherwise be intractable” [3].

References


A Appendix

A.1 Rewrite Rules

A catalogue of some of the core rules in the simplified top tier system is given in Figures 5, 6, 7, 8, 9 and 10.

Fig. 5. Request to Buy

Fig. 6. Begin Analysis

Fig. 7. Follow Result
A.2 Strategy Language Syntax

A summary of our chosen strategy language is given in Table 3. We refer to [7] for more details.
Let $L, R$ be port graphs; $M, N$ subgraphs of $R$; $W$ a subgraph of $L$;
$n \in \mathbb{N}$; $\pi_{1...n} \in [0, 1]$; $\sum_{i=1}^{n} \pi_i = 1$
Let attribute be an attribute label in $\nabla A$;
e a valid expression without variables

| Rules          | $T ::= L_W \Rightarrow R_M^N \mid (T \parallel T)$ |
|               | $\mid \text{ppick}(T_1, \pi_1, ..., T_n, \pi_n)$ |
| (Applications) | $A ::= \text{all}(T) \mid \text{one}(T)$ |
| (Focusing)    | $F ::= \text{crtgraph} \mid \text{crtPos} \mid \text{crtBan}$ |
|               | $\mid F \cup F \mid F \cap F \mid F \mid (F) \mid \emptyset$ |
|               | $\mid \text{ppick}(F_1, \pi_1, ..., F_n, \pi_n)$ |
|               | $\mid \text{property}(F, \sigma) \mid \text{ngb}(F, \sigma)$ |
| (Determine)   | $D ::= \text{all}(F) \mid \text{one}(F)$ |
| (Update)      | $U ::= \text{setPosition}(D) \mid \text{setBan}(D)$ |
|               | $\mid \text{update(functionparameters)}$ |
| (Properties)  | $\sigma ::= \text{Elem} \mid \text{Expr}$ |
|               | $\text{Elem} ::= \text{node} \mid \text{edge} \mid \text{port}$ |
|               | $\text{Expr} ::= \text{attributeRelope} \mid \text{true}$ |
|               | $\text{Relope} ::= == \mid != \mid > \mid <$ |
|               | $\mid \geq \mid \leq \mid \approx$ |
| (Comparison)  | $C ::= F = F \mid F \neq F \mid F \subset F$ |
|               | $\mid \text{match}(T) \mid \text{isempty}(F)$ |
| (Strategies)  | $S ::= \text{id} \mid \text{fail} \mid A \mid U \mid C \mid S; S$ |
|               | $\mid \text{if}(S)\text{then}(S)\text{else}(S)$ |
|               | $\mid \text{repeat}(S)[(n)]$ |
|               | $\mid \text{while}(S)[(n)]\text{do}(S)$ |
|               | $\mid \text{try}(S) \mid \text{not}(S) \mid (S)\text{orelse}(S)$ |

Table 3. Syntax of the Strategy Language.