Modelling an RTGS system with SLAPP

Claudia Biancotti
European Central Bank

Leandro D'Aurizio
Bank of Italy

Giuseppe Ilardi
Bank of Italy

Pietro Terna
University of Torino

February 10th 2009

Introduction

A payment system is the set of organisational procedures through which entities (individuals or institutions, either public or private) exchange and regulate their payments.

In a typical day of a developed country, the amount of economic transactions processed by the payment system is roughly 20% of the yearly GDP.

Real Time Gross Settlement systems have been implemented from the early nineties for major payments, and these payment systems are often directly managed by central banks.

In a RTGS system, a payment is settled only when the corresponding full amount is transferred across accounts held at a central bank, preventing systemic failures caused by major participants defaulting.

Motivation

Given the financial volumes at stake, central banks started in the last ten years to model RTGSs, with a special interest in understanding how to keep the structure stable and functioning even under extremely difficult conditions (Hellqvist and Koskinen, 2005).

In particular, the gross settlement eliminates the systemic credit risk but requires a huge amount of liquidity with respect to the DNS payment system. So the analysis of the liquidity flows is the critical aspect for a supervisor of the RTGS system (Central Banks).

As pointed out by Beyeler et al (2007) and Galbiati and Soramäki (2008) the treasurer's liquidity management game is an important point in this analysis and it heavily depends on the conditions for accessing to external liquidity funds.

The amount and the distribution of liquidity depend on complex interaction among system participants. Flows of liquidity are continuously exchanged and can be only partly predicted. This situation imply that it is difficult to define a statistical structural model for representing a real RTGS payment system.

Therefore, it is more convenient to model the relationship between necessary liquidity, delays and access to the monetary market at an agent level.

The agent-based approach, where complexity emerges from the single agents' interplays is particularly suitable in such a context, if the interest is understanding the underlying decisional processes and their consequences in real situations.

Related Literature

The previous literature can be roughly divided in two groups:

- the game-theoretic analysis, (Angelini (1998), Bech and Garratt (2003, 2006), Buckle and Campbell (2003), Willison (2005) and Galbiati e Soramaki (2008)) in which the analysis is focus on the "liquidity management games" and the use of incentives for obtaining an inefficiencies improvement;
- Realistic simulation analysis (Markose *et al.* (2006), Arciero *et al.* (2008), Kabadjova *et al.* (2008)) in which the authors investigate the consequences of alternative scenarios on payment delays, liquidity needs, and risks using actual payment data.

SLAPP

In this work, we develop a simulation tool for a RTGS payment system in a Agent-based Model perspective using SLAPP.

SLAPP is the acronyms for Swarm-Like Agent Protocol in Python, and it is a simplified application of the original Swarm protocol, choosing the Python language as a simultaneously simple and complete object-oriented framework.

Motivation of using SLAPP

In a previous analysis, Arciero *et al.* (JASSS 2009) presented a small-scale agent-based model of an RTGS system in StarLogo, with the aim of understanding the basic interactions and the aftermath of an unforeseen extreme event.

In order to overcome the intrinsic limitations of StarLogo, the current model is written in SLAPP and so it is able to deal with real payment data and to handle a large number of banks.

Swarm

The Swarm Protocol rigorously defines a structure for simulations, based on a discrete-event philosophy, where multiple agents are represented by an object-oriented representation.

Seminal concepts are those of the set of agents as a collection ("swarm") endowed with an activity schedule, as well as that of an observer running the model with a schedule to produce graphical representations, reports, etc.

The clock of the observer can be different from that of the model and this feature enables watching the simulation results with a flexible choice of the time frequency.

Swarm is therefore a set of procedures for defining agent-based simulation models, and this protocol is independent of the language in which we translate it.

Swarm has been written with the Python programming language, which creates all the model elements as instances of classes. In order to increase its usability, the python code is called by a R main routine

Our basic model

In the model, commercial banks are the agents that move between two worlds:

1. a market for short-term liquidity where agents exchange funds at a price represented by an interest rate;
2. a representation of a RTGS system.

In the current version of the model there is no link between the two world. In the next development, the bridge will be the amounts of liquidity chosen by each treasurer in each time.

At the beginning of each tick, the treasurer will choose the amount of payments to be settled taking in account the level of liquidity available and the expected future liquidity.

Money market

In the actual version of the model, agents enter the market as buyers or sellers according to a simple probability distribution chosen at the beginning of the simulation.

Two different behaviours can be simulated into the money market:

1. agents decide at what price buying/selling liquidity according to the most recent price quoted in the market;
2. the reference price is the most favourable one so far practised (respectively the lowest one if the agent is looking for liquidity, the highest one if the agent is selling liquidity).

RTGS

The second part of the model is the RTGS payments system. In this world, the bank's treasures (our agents) control the scheduling process for each payment. The aim of the bank is to settle the requested payments as soon as possible while at the same time controlling the amount of liquidity available. Hence, there is a clear trade-off between efficiency and liquidity beside the delays' strategic management.

An input dataset containing data of payment settled on a RTGS is provided to the model. The payment settlement time can be artificially put forward or back, in order to perform “what if” analyses under different distributions of payment delays. Agents can also by partitioned into two groups, the first composed by payers who do not tend to delay too much, the other containing those who pay later.

In a first exercise the system has been tested with payment data corresponding to an entire day of activity (more than 23000 transactions) in March 2008 on the Italian RTGS system, with around 50 banks regulating directly their payments (the system is tiered so that the smaller banks participate only indirectly in the system).

The following figure 1 shows the patterns of the cumulated amount of payments settled during the simulated day, together with the corresponding cumulated overall liquidity borrowed on the monetary market.

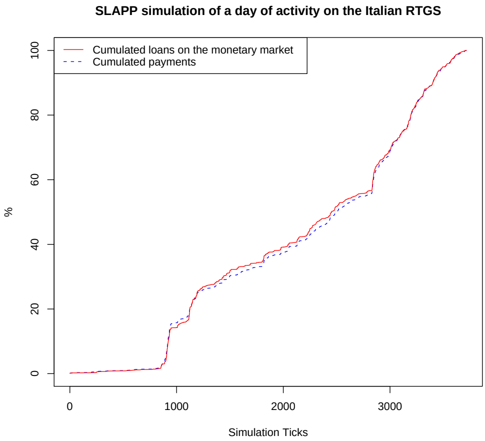


Figure: SLAPP simulation of a day of activity on the Italian RTGS system. Cumulated percentage amounts of payments (blue dashed line) and monetary market loans (red line).

The first indicator is a kind of throughput rate of the system (Markose *et al.*, 2006). In this simple scenario, the agents are settling payments and lending/borrowing liquidity with similar frequency, although we do not know what fraction of the payments are covered by loans. A heavy utilisation of the money market might be a clue of stress in the system: a possible indicator for that is the ratio between cumulated loan and payment amounts (see also Galbiati and Soramaki, 2008). This last indicator is shown in figure 2, where we can see that the percentage of the payment amounts settled through loans is quite stable around a mean value of 48.5 percent after a more variable pattern in the simulation burn-in.

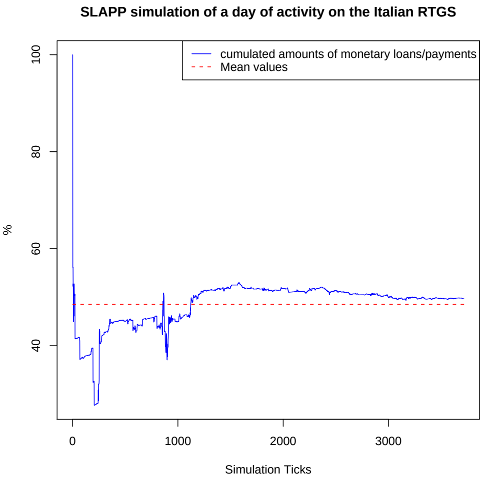
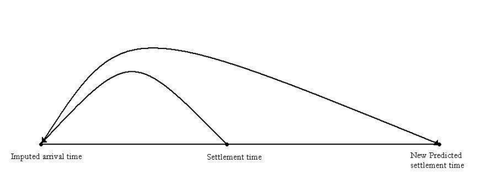


Figure: SLAPP simulation of a day of activity on the Italian RTGS system. Ratio between cumulated amounts of monetary market loans and payments.

A look at the mechanisms regulating payment internal queues

One of final aims of this work is to model the treasurer's behavioural rules so that, in future developments, we will be able to connect the settled payments to some factors such as incoming payment requests, needed and available liquidity and total amount of payments to be settled.

The key problem in this analysis is that we need the arrival time for each the payment settled. This information is usually missing, so we have to impute it by using a model estimated on an auxiliary dataset.



Some Stylized Facts

In this part, we show some aggregate evidence of the actual scheduling rules based on data collected from a leading Italian bank, for which the arrival time of payment request is available. The data correspond to three working days, selected so as to constitute a relatively homogeneous sample. For each payment included in the dataset, information on origin, settlement time, the amount and typology area is available.

In a typical day a treasurer usually manages a huge number payments of small size and a relative small number of large amounts.

However, these few transactions represent the bulk of the total amount, with the first quantile corresponding to 62% of the total volume, for instance.

The treasurers tend to stack the incoming payment requests and to release them once their number exceeds a certain threshold. Hence the major part of the requests is settled in few time-frames, rather than continuously in time. The next figure shows the time distribution of the number of settled payments in the three days.

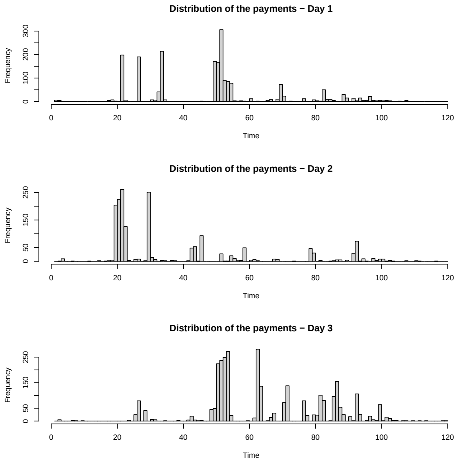


Figure: In the panels are plotted the numbers of payments settled in bucket of five minutes in three different days.

We can also gather from the same graph that a such queuing policy produces time-frames in which payments are released that are not evenly spaced during the day, since their actual time location can also depend on some payments that cannot be deferred. The next figure, featuring the time distribution of incoming and settled payments, provides a clearer insight of the decisional rule.

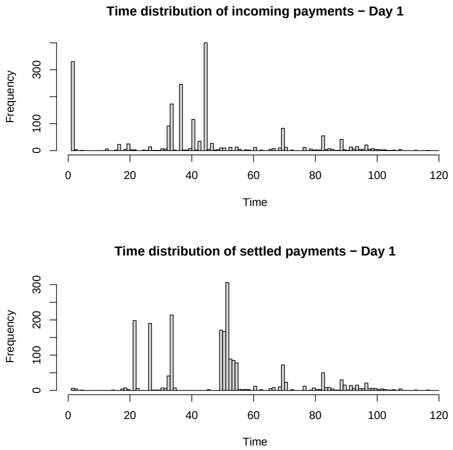


Figure: Top panel: time distribution of the payments entered in the internal queued. Bottom panel: time distribution of the payments settled. Time span 120 buckets of five minutes

The large part of payment requests arrives in the first part of the day and is also settled in large batches before the afternoon, without taking into account the consequence of possible delays.

On the contrary, payment requests arriving in the evening are settled more quickly, mainly because of the constraint that they should be settled before the 6 p.m. deadline.

If the simulation uses real payment requests, they also include their actual settlement time, whereas the time when the treasurer becomes aware of their existence is unknown.

In this preliminary version of the model this critical piece of information is therefore imputed by subtracting from the settlement time a random delay, drawn from an uniform distribution.

In the next version, the time of the arrival of the payment request will be generated more akin to the empirical evidence: the delay will be approximated by a finite mixture of truncated normal distribution.

This choice is justified by the natural ability of the finite mixture of adapting to multimodality, a peculiar characteristic of the empirical delay distribution.

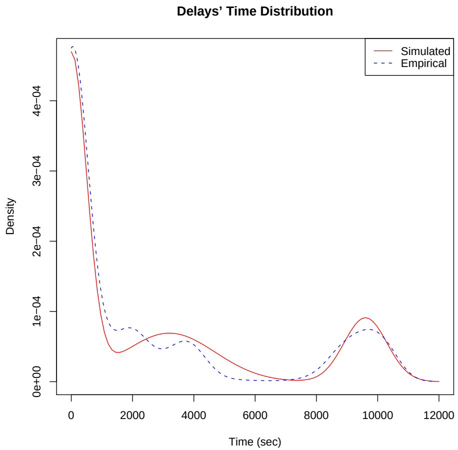


Figure: In the graph are plotted the empirical (blue dashed line) and the simulated (red line) time distribution of the delays.

Conclusions

The present paper illustrates the utilisation of agent-based modelling for the simulation of payment systems.

The main novelty of the approach is the application of the Swarm protocol through the Python language. The flexible programming structure of Python enables the user to create an easily upgradable and extendable model that is presently able to handle large-scale simulation scenarios with hundreds of banks and real streams of payment requests.

The paper also presents an investigation on the characteristic of the internal queuing scheduling with a particular emphasis to the reconstruction of the arrival time of the payment requests.