Integrity and Internal Control in Modern Banking Systems

Jim Jones
Havenlogic Systems Ltd.
7 Oakcroft Close, West Byfleet, Surrey
UK, KT14 6JQ
Tel.: +44-1932-355716, e-mail: jim-jones@havenlogic.fsnet.co.uk

Abstract: This paper examines the nature of controls traditionally applied in the Banking industry to batch transactions. It then looks at the reasons why batch systems introduce the notion of Risk into financial systems and why modern systems are moving towards real-time in order to overcome this Risk. By analysing real-time banking systems, including those driven by the advent of the Internet, the paper identifies the need for an entirely new family of controls and proposes an architecture of Parallel, Autonomous Audit as a framework in which this might be developed.

Key words: Batch controls; real-time controls; distributed transactions; Parallel, Autonomous Audit

1. INTRODUCTION

"You may not like your bank but you do trust them"

This comment constituted the principal theme of the SIBOS banking conference in San Francisco in October 2000 where the major banks of the world started to come to grips with the risks inherent in using the Internet as a banking medium.

The business of banking has always focussed very heavily on the need for Controls. Double Entry book-keeping itself is a form of control; hash

M. Gertz et al. (eds.), *Integrity, Internal Control and Security in Information Systems*

totals and batch totals are forms of control; transaction reconciliations are forms of control.

Despite this focus, transactions continue to go missing or end up at the wrong destination and in the worst case, banks continue to go bankrupt. Therefore, there still remains a need to enhance controls for the systems we have been familiar with for many years. However, it is clear that we are also being faced by a new generation of banking system and, if our traditional controls are not adequate for the old systems then it must be questioned to what extent they are appropriate for the new systems.

In this paper, it is proposed that these 'traditional' forms of control are no longer adequate for the new generation of banking systems which can be categorised as real-time, distributed transaction systems. We will explore the architecture of a number of these systems, including Real-Time Gross Settlement (RTGS), Continuous Linked settlement (CLS), Straight-Through Processing (STP) as well as Internet banking. We will try to establish what makes these systems different, what new challenges they pose for control regimes and some ideas for ways in which these new risks may be addressed.

2. TRADITIONAL BANKING SYSTEMS

Since banking was systematised by the Florentine bankers in the 15th century, the business of banking can be crystallised as 'the buying and selling of money'. A customer establishes a legal relationship with a bank via one or more bank accounts; in order to come into operation, these bank accounts need to be funded (hence money is paid into them) and they operate by means of the input and output of funds. In some cases, the transfer of funds is an end in itself; for example, if you go to an automated teller machine (ATM) to draw cash from your own bank account, the transfer of funds is purely a financial one. In other cases, however, the movement of funds is used to support some other type of business transaction (often referred to as an 'underlying transaction') such as the sale of purchase of stocks and shares or the repayment of a mortgage or the receipt of interest on a savings account.

These systems are transaction-orientated and are collectively referred to as Payment Systems. Indeed, there is a view that Payments are not restricted to banks but extend to other financial organisations, such as Credit Card companies (such as Visa), Building Societies, Mutual Funds etc.

In traditional systems, the processing of Payments was viewed as a routine task conducted in the back office of a bank. Transactions were submitted on paper (cheques being but one example) and each transaction was hand-written into a large Ledger book. This manual process took some time to perform, especially as any checks on the validity of the transaction, such as credit checking, were also manual. However, the banks were not unhappy about this situation, since they were able to make use of this money (a float) for several days and earn interest on it until the transaction was finalised or settled and they relinquished the money.

From the mid-1960s onwards, the introduction of computer systems began to impact some of these fundamental assumptions about the way Banking should work. For example, it was found that computers worked best by processing a group of transactions together as a batch or file, which meant that, instead of each transaction being handled individually as in the manual system, transactions were accumulated until a large enough set was available for computer processing. Although the computer processed these batches of transactions far quicker than a human being could, nevertheless, there was still a considerable period of time in which a transaction simply waited to be processed.

Batch computer systems found it easy to adopt traditional banking controls. Batches were totalled at various stages of processing and the totals checked to ensure that a transaction had not been corrupted, either accidentally or (in some cases) intentionally. At the end of the processing period – usually a day – a summary or statement would be printed and reconciled back to the initial transactions.

The prime characteristic of these systems was that, even though it might still take several days to complete the processing of a transaction (in the UK, today, it still takes three working days for most payments), the interim checks and balances enabled errors to be picked up and corrected, long before the transaction was finalised and certainly long before the customer could become aware that a problem had occurred. As a result, banks were perceived to handle the overwhelming proportion of their transactions correctly and customers felt comfortable that they did so.

3. THE AGE OF INNOCENCE VANISHES !

In the 1970s, the major banks started to realise that all was not well with their systems. Customers were beginning to demand quicker responses, based on the fact that banks were spending lots of money on computers, and the systems began to creak. In addition, there was a sudden realisation that Payments were not merely an administrative function in the back office but, if handled incorrectly, would have severe impacts on the bank's ability to fulfill its obligations. In simple terms, if a bank tried to pay out all its obligations before it received any incoming money, it would lack the liquidity or available cash to do so and would stop functioning.

It therefore became clear that there is not only a relationship between incoming and outgoing payments but also a complex relationship between the timings of them. In 1930, the major banks of the world established the Bank of International Settlement (BIS) in Basle, Switzerland, to research the impacts of various flows of Payments and recommend best practices to safeguard these. Throughout the 1980s and 1990s, the BIS published a series of recommendations to the Banking industry. In cases where the commercial banks were slow to implement these recommendations, the Central Bank in a country would intervene and impose a much more onerous solution, typically requiring the commercial bank to submit large amounts of collateral to guarantee that transactions would complete. Of course, such collateral could not be invested and the banks lost interest. Hence, the commercial banks were incentivised to implement BIS recommendations.

4. REAL-TIME GROSS SETTLEMENT (RTGS) – THE FIRST NEW SYSTEM

Even though computers had reduced the time needed to process a Payment, it still took at least one working day before a transaction was finalised. During this period, therwas a risk that something could go wrong. It might be that the customer who had asked for the transaction went bankrupt; it might be that the bank itself lacked liquidity and went bankrupt.

Therefore BIS established a two-tier classification of Payments – High-Value and Low-Value. Although each country is free to determine the boundary between these two types, in practice, most countries come to similar conclusions. In the USA, the Federal Reserve classifies a payment over US$ 1 million as a High-Value payment; in the UK, the Bank of England classifies a payment over UK£ 1 million as High-Value.

In an RTGS, the paying customer (usually a Corporation or a Government department or, in fact, a bank) issues an instruction to his bank to make a high value payment on a certain day. Usually, high value payments are not triggered on impulse; they are scheduled several days or even weeks in advance. Therefore, on receipt of the instruction, the bank stores it in some form of data warehouse and, during this storage period, the customer is free to cancel his instruction

On the due date, the bank extracts the instruction, checks that the customer still has funds to cover the payment and, if all is correct, sends the instruction to the national bank of its country for immediate finalisation (usually called Settlement). As soon as the instruction is settled, the beneficiary's bank is notified and for practical purposes the funds are available to the beneficiary.

The hypothesis is that a High-Value payment is too important to be left in limbo for a day before it is finalised and the BIS therefore recommended that it should be processed by computer in real-time. The idea was that even if 'real-time' in reality meant 15 minutes, this was substantially better than 8 or 10 hours or worse.

The corollary of this idea, however, was that batch controls were inadequate to safeguard such transactions. In a real-time system, clearly there is no concept of batch controls and so, this weapon was lost. Likewise, if it took from 9.15 in the morning to 9.30 to process a High-Value payment, it was of little use finding out at 5.00 in the evening that it had been processed incorrectly – the money was gone and it was difficult, if not impossible, to get it back.

What was needed was a set of controls, which would monitor each individual transaction, and alert the bank to any error in processing before the transaction was completed.

5. DISTRIBUTED PAYMENTS

In 1974, the Banking industry was rocked by the bankruptcy of Bankhaus Herstatt in Germany.

This bank was processing foreign exchange transactions and a situation arose where it was buying Japanese Yen and selling US dollars. Due to the time differences around the world, it paid out to buy the Yen early in the day while the Tokyo market was awake but had to wait for the US market later in the day before receiving the incoming payment. Unfortunately, when the US market awoke, there were no dollars and Bankhaus Herstatt could not sustain the loss and was declared bankrupt.

This risk – the risk arising from different trading hours around the world became known as Herstatt Risk and the BIS set out to address the problem. This led to a set of recommendations in 1995 and ultimately to the development of a system to solve the problem, known as Continuous Linked Settlement (CLS).

You can look upon CLS as a two-sided RTGS, described earlier. Imagine a typical high value foreign exchange transaction: a major US motor manufacturer wishes to pay for a new factory in Switzerland. So, it sends an instruction to its bank – please transfer the equivalent of US$ 500 million in Swiss francs into our account in Zurich at 10.00 next Wednesday.

The bank therefore has to sell US dollars and buy Swiss francs which it does through it foreign exchange trading department. The usual way in which this is done is for the trader to find another bank which wants to convert Swiss francs to US dollars for its customer and the two banks basically agree an exchange of value.

Again, the transactions are scheduled well in advance and so each bank sends its half of the transaction, in this case, to a data warehouse at CLS bank. On the due date, CLS Bank looks at both halves of the transactions and if they match, it settles the two transactions simultaneously. If either side of the transaction is missing, settlement does not occur and the half of the transaction which was correct is returned, with no money being lost. The same thing happens if both sides of the transaction are present but do not agree in value.

In this way, the CLS system overcomes the Herstatt problem by insisting that both sides of the transaction are available simultaneously.

CLS provides a number of control features, not all of which will be described in this paper. Some controls are the traditional end of day controls, which are the ultimate check that everything has been processed correctly. Other controls relate to the real-time aspects of the system, the mechanics of which have been described above.

However, the study of 'how foreign exchange transactions work' has led to the identification of another issue, the importance if which is just beginning to be realised.

The architecture of the CLS system is illustrated below.

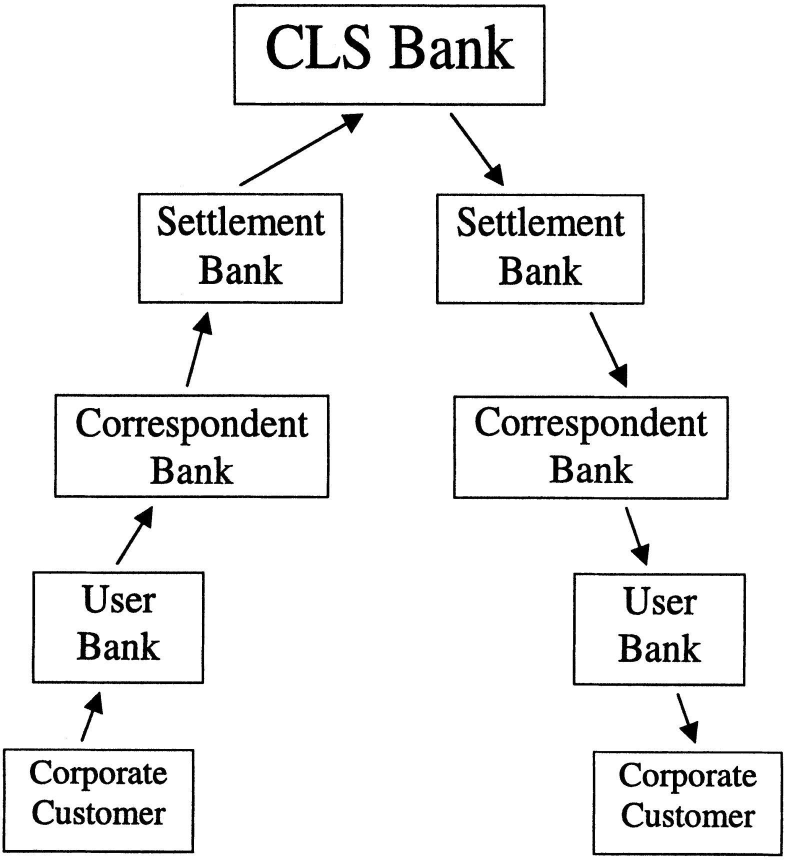

Figure 1. CLS System Architecture

What we can see is that, from the time the transaction is initiated by the customer to the time that the money arrives in the recipient's bank account, the transaction passes through a number of separate organisations, usually banks, on its route to its destination. Each bank is legally only responsible for the transaction for part of its life and there is no single organisation, which has the overall authority and mandate to monitor the integrity of the transaction for the whole of its life cycle.

The question therefore arises – who is supposed to do what if a transaction goes wrong ?

Clearly, any error should be picked up at the end of the day when CLS Bank issues its statements for the participants to carry out their reconciliations. If a participating bank identifies an error, it first has to check its own systems to see if it is the culprit, but if it can find nothing wrong, then it has to pass the problem to the next bank in the chain for it to carry out an investigation and so on. If nobody will admit to having caused the problem and they all point fingers at each other, the resolution could take several days. However, as has been pointed out earlier with RTGS, the actual money is long gone and is difficult to recover. If one of the major banks participating in this system considers a typical transaction to be US$ 500 million, the misprocessing of a single transaction is a serious event not only for the bank involved but also the other banks who could be affected by a knock-on or systemic effect.

Finally, as far as the customer is concerned, he neither knows nor cares how many banks are involved in the processing chain. He requires his own bank to fix the problem, regardless of whether they caused it or not.

6. DISTRIBUTED PAYMENTS – ANOTHER EXAMPLE

CLS is by no means the only new Banking system to adhere to the architecture described above and to suffer from the same risks. Within the Securities market, there exists a similar problem. As shown below, in the buying and selling of Securities, there are a number of organisations involved. Some of them still have manual systems and those that are automated have incompatible systems.

Historically, this has meant that once one system has processed a transaction, it prints out the results and passes the paper record to the next organisation in the chain, which re-enters the details, obviously with the risk of making an error.

To address this, the organisations involved in Securities processing are in the act of launching initiatives to eliminate these manual steps into what is becoming known as Straight-Through Processing (STP). The key idea here is that all the computers will speak a common 'Securities language' with common message formats, so that transactions can be passed automatically from one system to the next without manual intervention.

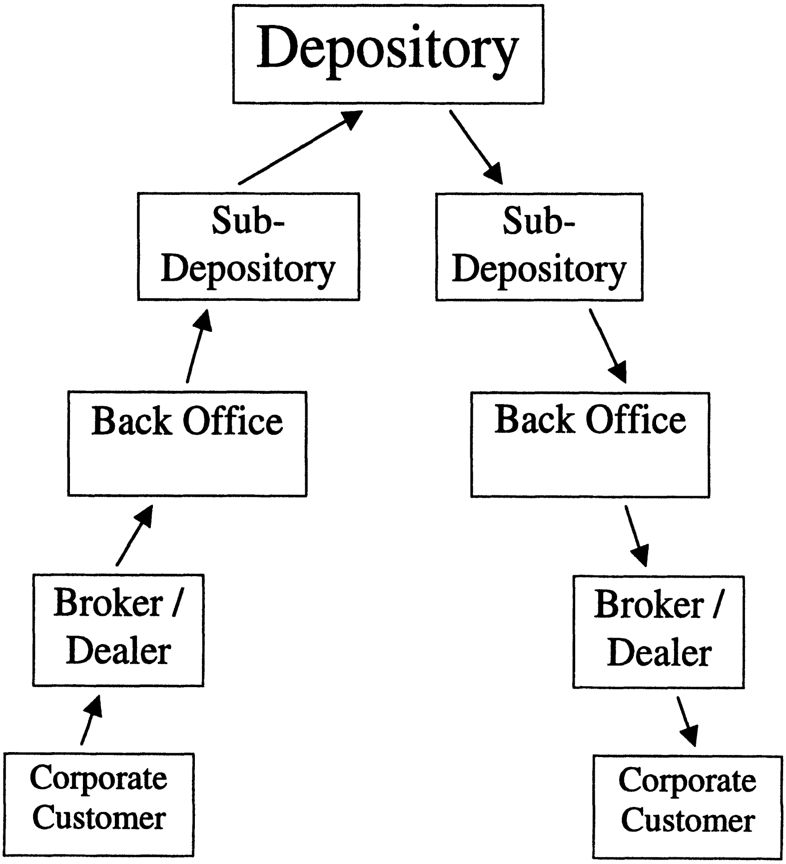

Figure 2. STP System Architecture

Taking a typical transaction, if you instruct your investment broker to buy some shares in a company for you, he looks around to find another broker whose customer wants to sell those shares and when he finds one, the two brokers agree a deal. Today, each broker handwrites on a piece of paper the details of what he thinks he has agreed over the telephone and a messenger takes this piece of paper into the back office.

In the back office, the details are keyed into a computer system and transmitted to the other broker's computer system. Even at this early stage, there is room for error. Handwritten notes are transcribed incorrectly; pieces of paper become detached and fall on the floor or they simply lie on someone's desk unnoticed.

Once past this stage, the transaction moves from the back office to a custodian (which is usually a bank which holds the customer's share certificate) up to a Depository where the transfer of ownership is registered and then all the way down the other side of the pyramid.

The other problem with this type of system is one shared with CLS. None of the organisations in the chain is responsible for the transaction for the whole of its life cycle which means that there is no clear allocation of responsibilities for fixing anything which goes wrong.

One possible solution to this problem is to have a neutral 'referee' sitting outside the system but monitoring the transaction as it passes from one organisation to the next and it has been suggested that SWIFT might play this role. The basis for this is that SWIFT has defined a set of standard message formats which handle securities transactions, SWIFT provides the network which links together the banks which process the corresponding payments and SWIFT has ambitions to use the same network to link together securities firms on a worldwide basis.

It is encouraging to see that this type of control issue (which we look upon as decentralised control) is being addressed but the SWIFT solution is, at best, a partial one. Firstly, many of the larger securities firms have already implemented their own version of STP with their own networks and it could be disruptive to replace this with SWIFT. Secondly, SWIFT can only monitor what goes into an organisation and what comes out of the other side. As such, SWIFT may identify which organisation is causing a problem but cannot tell which system within the organisation is at fault. This paper explores below an alternative approach towards handling decentralised control at a more detailed level.

The elimination of error-prone manual processes is clearly a good thing but, by itself, does not eliminate all the new problems we have identified to date:

- As processing is automated and moves nearer to real-time, traditional end of day controls no longer pick up errors early enough
- In a system made up of autonomous organisations, there is no overall control of a transaction from beginning to end
- Even worse, a Securities transaction is one which we have earlier classified as an underlying transaction; in other words, it involves the buying and selling of equities and the transfer of ownership. However, this transaction is usually accompanied by some form of payment, which is going along in parallel but separately. If the two transactions are not

synchronized, there exists the possibility that either you have received the equities but have not paid for them or, possibly worse, you have paid for them but you are not yet the legal owner !

Hence, you have two systems, each with its own risks, trying to work together, without creating additional risk (this risk is sometimes referred to as Delivery versus Payment)

Hence, this type of system incarnates all of the risks described above and needs to address them all.

7. DISTRIBUTED PAYMENTS – A FINAL EXAMPLE

In the introduction to this paper, we referred to the issue of Internet banking, which caused so much stir at Sibos.

Internet banking is in its infancy and means different things to different banks. At one extreme, the Internet is simply another wire to connect the customer to his bank. The types of transactions, which are carried out, are the same as before and can be controlled in the same way. However, some major banks have ambitions to use the Internet to provide new services. One of these – the concept of acting as an Exchange – is illustrated in figure 3.

The banks observe that the Internet is allowing new organisations to set up so-called Exchanges to bring together Buyers and Sellers in a variety of ways. One typical example is to organise electronic auctions in which the Buyer hopefully ends up with a lower price. Another is the electronic organisation of Tendering – publishing Invitations to Tender (ITTs), establishing bidding consortia, managing the bidding process and managing the contracting process.

Why should the banks want to get involved in this type of business process ? Two reasons are commonly quoted:

- In the life cycle of a total business transaction, the payment (i.e. the part the bank has traditionally handled) is only a small part. If the exchange decides to offer payment services, the bank is cut out of the business completely —it is disintermediated.
- Whoever runs an exchange function learns a huge amount about both the Buyers and Sellers and is therefore well equipped to offer additional services. Banks have always suffered from a lack of understanding of

their customers' habits. Whereas a retail store knows all about the food you buy and the clothes you wear, as a simple by-product of your purchases, and can predict what you are likely to buy in the future, a bank has to explicitly ask for this information and such requests often meet with resistance.

There are many risks involved with a bank going into these areas of its customers business and a bank has to weigh up very carefully what new legal liabilities it will incur with each new service it offers. For example, if it sets up an auction for a buyer and selects a seller, what is its legal liability if the seller disappears with the money but without delivering the goods.

It is not possible in this paper to evaluate all of the risks involved in this new and evolving type of trading, but we can see that all of the risks described earlier apply to this area:

- Transactions are moving to real-time
- A number of organisations (NOT all of them banks) are involved in the chain
- There is an underlying transaction – usually a purchase – which has to be synchronised with the Payment.

However, in addition, it is useful to highlight an additional risk (which does occur in some of the earlier scenarios but which we have chosen to hold off describing up to now). This is the risk associated with a long transaction.

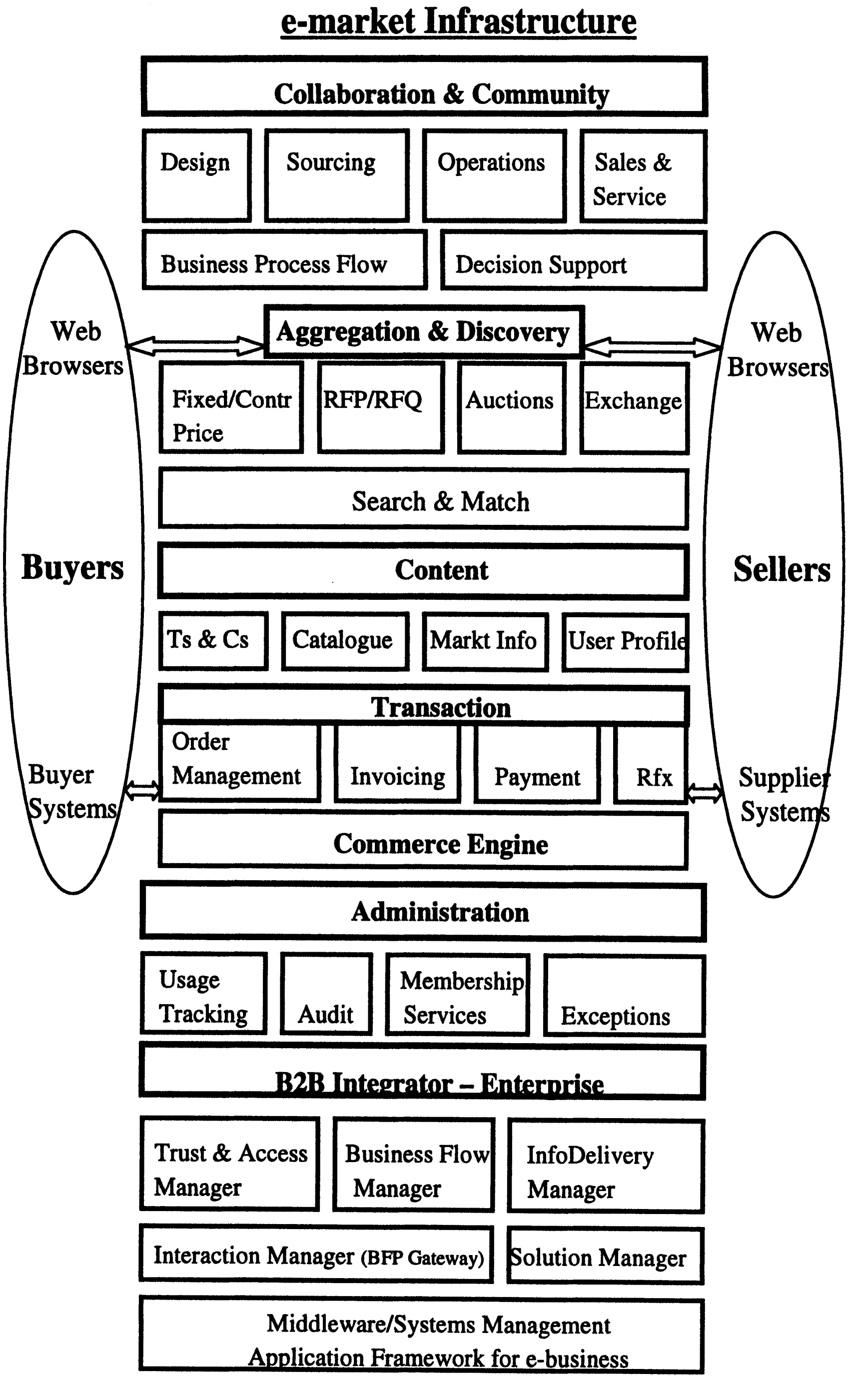

Figure 3. Electronic Exchange Architecture

8. LONG TRANSACTIONS

We have seen that banking transactions now can be processed in a few minutes up to one working day and other parts of the financial industry, such as Securities, is working towards the same position. Nevertheless, there is a type of transaction which is intrinsically different; this is a transaction which is intended to last more than one processing day.

A typical example in a bank or broker is a Futures transaction where the two parties agree to conduct a transaction at some future date. The transaction is recorded immediately but may not be completed for up to one year – and, in fact, may never be completed but cancelled after a year.

In the case of Internet banking, especially in the sector known as Business to Business, if the object of a bid is a major capital item, then payment is typically staged over a period of time and is triggered by certain conditions, such as the completion of a stage of the work or a partial delivery.

The common point is that a transaction, once recorded, lasts for a considerable duration. Of course, such transactions have always occurred and they occur in many industries other than Finance. However, the size of the transaction and therefore the size of the payment involved and its impact on the bank if it is not handled correctly, means that it is not enough simply to record a transaction on a magnetic medium and recall it on the due date: it is necessary to monitor it all the time it is dormant to ensure that if anyone (either within the bank or from the customer) tries to access and change that transaction, they have the appropriate authorisations.

9. SUMMARY OF THE NEW REQUIREMENTS

In the new generation of Banking systems, we have seen that there is a set of new risks, which requires a new set of solutions:

- A need to monitor a real-time transaction during its life cycle
- A need to establish a control regime where ownership of a transaction is distributed over a number of autonomous organisations
- A need to ensure that such systems are properly synchronised
- A need to monitor long transactions

10. TOWARDS A SOLUTIONS ARCHITECTURE

For each of the four categories of new risk listed in section 9, an individual solution either already exists or can be conceived.

10.1 Real-time transactions

There is a need of a software monitor which also runs in real-time and maintains a set of business rules which define key moments in the life of the transaction which need to be reported on and checked – with a corresponding real-time warning or other action

10.2 Distributed control

There are two types of solutions here, which could be used independently or together. Collectively, they may be referred to as Transaction-Orientated Life Cycle Audit Trails (TOLCATs) because they are founded on the idea of detaching an audit trail from a computer system and, instead, attaching it to a transaction.

The first is to take the concept of double entry book-keeping and apply it to a transaction. This could mean that you initiate a transaction on its route through the transaction chain and then you send off a duplicate of this transaction around either the same or an alternative network and you match up the two versions of the transaction at certain intermediate points in order to highlight a problem as early as possible in the cycle.

It might be objected that this doubles the processing required. However, if we recall that this type of processing applies only to High-Value payments (of which there are comparatively few each day), then this could be considered as an acceptable insurance. A more powerful objection might be – what do you do if both transactions do not arrive at a checkpoint at the same time ? Do you wait (and, if so, how long) for the second version which may never arrive ? And, if the two versions do not match, which one do you take to be correct ?

A second approach is to have the financial transaction carry its own control information, as a snail carries its house on its back. On reaching an interim destination, the transaction unpacks the control information, checks itself and if correct, carries on to the next point. This technology already exists on the Internet and is relatively platform independent. If you are browsing the web and suddenly a special offer appears on your PC screen which you had not asked for, in effect what has happened is that some organisation has downloaded some software onto your PC and set it running to display the offer. In current jargon, such software downloads are known as Non-Persisting Cookies !!

In the sort of chained transaction life cycle we have described above in CLS and STP, the issue is that you can only have a central monitoring system if everyone in the chain agrees to play and lay their systems open to the monitoring station. If only one participant decides against this (possibly because it involves purchasing hardware and software from some vendor it does not otherwise deal with), then the centralised approach breaks down.

A TOLCAT approach therefore appears to be the only feasible alternative in which the control information is implemented in a cookie, which accompanies the transaction. Obviously, the cookie has to be implemented in such a way that it is not trapped as an error by a firewall and to this extent, all participants in the chain have to agree to this.

If, however, someone objects to the idea of a piece of software loading itself into his computer and executing itself, then the community has to accept that it will not be possible to monitor the transaction as closely as they think necessary.

If we look at the practicalities, quite apart from the Internet cookie described above which nobody violently objects to, already one software house is using this technique to monitor Electronic Bill Presentment and Payment – a financial system similar to those described above which we did not have space to analyse. Moreover, System Management tools such as IBM's Tivoli series use the same techniques to check that remote computers are working properly, whether they belong to the one organisation or not.

It may therefore be hoped that this technique will achieve de facto acceptance in Banking and Finance scenarios.

10.3 Synchronisation of systems

Synchronising a Securities trading system with a payments system is usually done by both organisations in collaboration. However, such is the critical nature of the synchronisation and the consequences if it fails that it appears justified implementing an autonomous control to check if it has been correctly effected. Software is needed to sit on top of the interface to detect errors and arbitrate over whose software caused the problem

10.4 Long transactions

Here there is a need to monitor a transaction as though it were a piece of static data, such as a name and address and to maintain an audit trail of who accessed it under what authorisation during its extended life cycle.

11. CONCLUSIONS

From the four focus areas discussed above (RTGS, CLS, STP, Internet or e-banking), we can see that these systems are compelling us to look at controls in a number of new ways

- We have to take account of real-time transactions and the fact that traditional batch or end of period controls cannot report errors in these quickly enough.

- We have to take account of the fact that financial transactions have a life cycle which extends well beyond the boundaries of any single participant.

- We have to take account of the fact that transactions can no longer be seen as transient events which simply update master file information. They have an extended duration and need to be protected just as carefully as any other persistent data.

These characteristics lead us to conclude that they cannot be addressed solely by tinkering with existing controls. A radical new approach is required and this new approach requires a new control architecture. This architecture has several dimensions, as illustrated in figure 4.

Firstly, we stress that the architecture needs to be autonomous. This means that the control functions are independent of the applications. In handling high value payments, it is essential, in order to reduce the possibility of fraud, that the controls are managed by an entity which is independent of the application developers. It is clearly wrong to allow developers to develop their own controls.

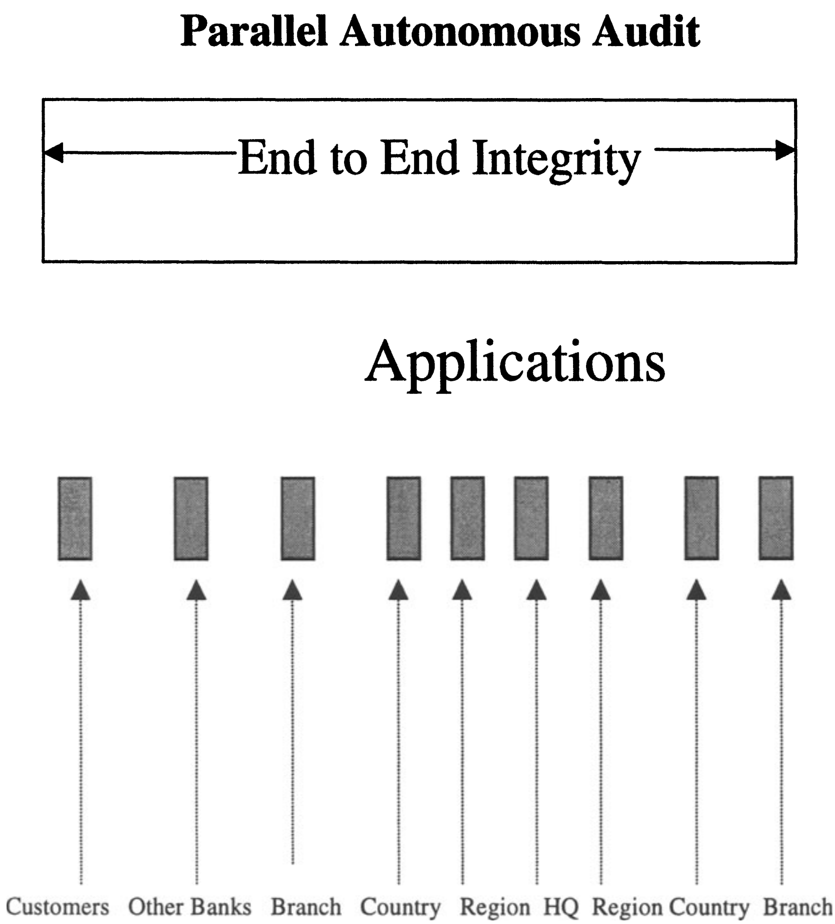

Figure 4. Parallel Autonomous Architecture

Additionally, in real-time systems, since we assume that the new controls will be developed in software, it is logical to suppose that control software may also have bugs. If such control software is in-line with the application code, it becomes part of the problem; whereas, if it is separate, then it can be turned off in the case of problems without affecting the on-going operation.

Secondly, the controls need to run parallel to the transactions they are controlling. If the transactions are occurring in real-time, the controls have to monitor them in real-time. If an error is detected, it needs to be reported in real-time so that the appropriate correction can take place, while limiting the propagation of the error.

Thirdly, the control regime needs to take account of the total life cycle of the transaction:

- In space – as it travels through a variety of autonomous organisations
- In time – for the duration of its existence, no matter how many years this may represent.

This paper therefore proposes a radically new paradigm for Controls in the Banking and Finance Industry. However, it is true to say that the factors which influence this and the solutions proposed may be applied to many other industries. One could think of airline scheduling, process control in steel mills, car assembly lines, drug prescriptions and a host of others.

12. REFERENCES

Valdez S (1993): An Introduction to Western Financial Markets Macmillan Press

Marshall C (2001): Measuring and Managing Operational Risks in Financial Institutions John Wiley & Sons (Asia) Pte Ltd

Bank For International Settlements (1989): Report on Netting Schemes G10 Committee on Payment and Settlement Systems

Bank For International Settlements (1990): Report on Interbank Netting Schemes (Lamfalussy Report) G10 Committee on Payment and Settlement Systems

Bank For International Settlements (1992): Report on Delivery versus Payment in Securities Settlement Systems (DVP Report) G10 Committee on Payment and Settlement Systems

Bank For International Settlements (1993): Report on Central Bank Payment AND Settlement Services with Respect to Cross-Border and Multi-Currency Transactions (Noel Report) G10 Committee on Payment and Settlement Systems

Governors of the Central Banks of the European Union (1994): Minimum Common Features for Domestic Payment Systems (Padoa-Schioppa Report)

Bank For International Settlements (1995): Report on Cross-Border Securities Settlements G10 Committee on Payment and Settlement Systems

Bank For International Settlements (1996): Report on Settlement Risk in Foreign Exchange Transactions (Allsop Report) G10 Committee on Payment and Settlement Systems

Bank For International Settlements (1997): Report on Real-Time Gross Settlement Systems G10 Committee on Payment and Settlement Systems