IJAREEIE

e-ISSN: 2278-8875
p-ISSN: 2320-3765

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

Volume 10, Issue 1, January 2021

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.122

9940 572 462 6381 907 438 ijareeie@gmail.com www.ijareeie.com

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 7.122|

||Volume 10, Issue 1, January 2021||

DOI:10.15662/IJAREEIE.2021.1001002

Applying the Transaction Management Techniques used in Distributed Heterogeneous Systems to Electronic Funds Transfer Networks

Elias Munshya[1], Simon Tembo[2]

MEng ICT Student, Department of Electrical & Electronics, University of Zambia[1]

Department of Electrical & Electronics, School of Engineering, University of Zambia[2]

ABSTRACT: In Zambia, banks have installed automated teller machines (ATMs) at strategic locations to enable their customers to have access to cash and carry out other banking services. Most of these ATMs are interoperable enabling customers from other banks to use them. One common problem when carrying out an interbank ATM cash withdrawal transaction is the customer's account being debited but the machine not dispensing the money. This problem is not resolved automatically in most cases, but instead, a customer is required to fill in a complaint form so that this issue can be investigated between the participating banks.

Before the switching of ATM transactions was migrating to Zambia’s national financial switch, the expected resolution time for this type of error was 42 days. Nowadays the period has reduced to significantly to around 14 days. This failure is still a great inconvenience to the customer. This research investigated this problem in Zambia.

KEYWORDS: Distributed Heterogeneous systems, two-phase commit protocol, transaction reversal fraud, ISO 8583, transaction reversal fraud.

I. INTRODUCTION

In Zambia, access to financial services has been improving with time. Many providers have introduced several innovative solutions to enhance and improve access to digital financial services by their customers. Digital Financial Services (DFS) have the potential to upend the traditional banking structure [1].

Unlike in the developed countries where digitalization has evolved significantly, African countries like Zambia still heavily rely on paper money to carry out transactions [2]. To enhance service delivery to customers, many banks have installed ATMs at strategic locations. Nowadays, a customer can access more services from an ATM, unlike in the past where ATMs only served the purpose of issuing money and checking the account balance. Today, a user can deposit money, pay utility bills, and even carry out cardless ATM transactions. Interoperability has also been achieved such that customers can access an ATM of any bank provided they are in the same scheme.

Database management systems define a transaction as a logical unit of work. This principle is used to group various operations on a database to execute as a logical unit in which all operations should either execute or non-execute at all. Given a transaction T consisting of a set of operations T = {O1, O2, O3, Ot}, all the operations in T are supposed to execute or non-execute at all. If O1, O2, O3 succeed, but Ot fails, all the other three successful operations must roll back any changes made. This property is called atomicity.

In an ideal world, it would be desirable that during the process of carrying out an EFT transaction likewithdrawing money from an ATM, any technical challenges that may be encountered which prevent the transaction from executing successfully to completion, should result in the reversal of any operations which have already executed. This is not always the case for interbank ATM transactions, as a manual reversal would have to be initiated. This breaks away from the true definition of a transaction.

Management of transactions in distributed heterogeneous database systems (DHDS) has been researched extensively, and a lot of literature is now available. DHDS and EFT’s are very similar in architecture as they both comprise heterogeneous nodes that need some interoperability to carry out specific functions. One protocol that has been used to

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 7.122|

||Volume 10, Issue 1, January 2021||

DOI:10.15662/IJAREEIE.2021.1001002

ensure atomicity is the two-phase commit protocol (2PC). This research critically looked at this problem and looked at the possibility of using 2PC to ensure the atomicity of the ATM cash withdrawals.

II. RESEARCH OBJECTIVES

The following were the objectives of this research:

I. Investigate why failed interbank ATM transactions cannot be reversed automatically in Zambia.
II. Investigate if the two-phase commit used in distributed heterogeneous database systems to ensure that transactions execute atomically can also be used in the ATM payment network schemes to ensure the atomic execution of the transactions.
III. Develop and test a transaction state incrementing algorithm to handle transaction errors during the processes of carrying out an interbank ATM withdrawal.

III. LITERATURE REVIEW

TRANSACTION MANAGEMENT AND PAYMENT ECOSYSTEM OVERVIEW

A. Interoperability

Interoperability is the ability of different systems, devices, applications, or products to connect and communicate in a coordinated way, without effort from the end-user [3]. At the functional level, different systems, regardless of developers, should access and transmit data between them.

Interoperability can be achieved in two ways, semantic interoperability, and cross-domain interoperability. Semantic interoperability is the type of interoperability in which two or more systems use common data formats and communication protocols to communicate. Common data formats and protocols which can be used in communication include XML and SQL. Lower-level data formats also contribute to syntactic interoperability, ensuring that alphabetical characters are stored in the same ASCII or a Unicode format in all the communicating systems [4]. Cross-domain interoperability, on the other hand, refers to the standardization of practices, policies, foundations, and requirements of disparate systems. Rather than relating to the mechanisms behind data exchange, this type only focuses on the non-technical aspects of an interoperable organization.

B. Payment Switch

A payment switch is a software that enables interoperability between different payment providers through the facilitation of communication between them. Switch typically provides a merchant-driven rules-based authorization and switching solution [6].

It routes payment transactions between several participants in the payment processing chain which include, acquirers, payment service providers, merchants, etc. This results in the extension of the payment network by easily adding providers without much effort.

Zambia officially launched its national payment switch on the 19th of June, 2019. Before the payment switch launch, all domestic interbank ATM transactions were being switched from outside the country through the VISA payment network. After the launch, all domestic ATM transactions were switched locally [6].

Later the switch started switching mobile money payments, which facilitated interoperability among different mobile payment operators in Zambia. This enabled customers to send money to any mobile money user regardless of the network service provider and send and receive money directly from bank accounts.

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (IJAREEIE)

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 7.122|

||Volume 10, Issue 1, January 2021||

DOI:10.15662/IJAREEIE.2021.1001002

C. ATM Transaction Processing

The processing of an interbank ATM cash transaction withdrawal is depicted in figure 1.

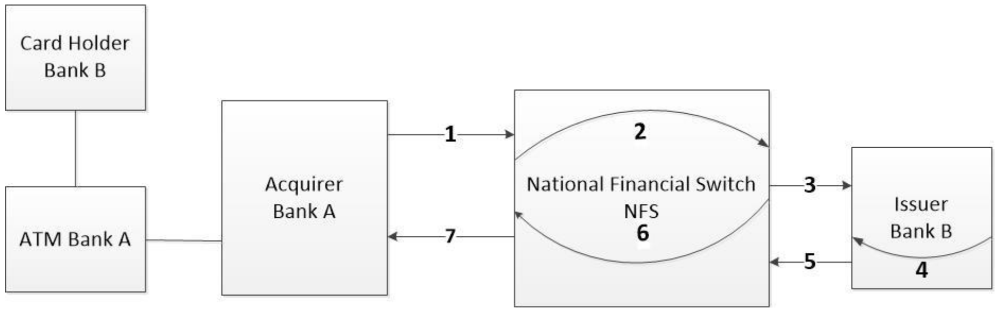

Figure 1: ATM Processing

1. Bank A will send an ISO 8583 transaction message to NFS
2. NFS routes the message to bank B channel
3. Bank B channel will route the message to Bank B
4. Bank B will authorize based PAN and PIN and debit the respective account.
5. The bank will send a response to the NFS regarding the transaction status
6. The ISO message will be forwarded to Bank A
7. Bank B channel will then forward the message to the bank, and cash dispensed

The ATMs use the single messaging system in which authorization, clearing, and settlement occur in a single message.

D. Distributed Database Systems

A distributed database is a database in which data is stored across different physical locations [7]. It may be stored in multiple computers located in the same physical area or may be dispersed over a network of interconnected computers.

Distributed databases are generally classified according to their environment into a homogeneous and heterogeneous distributed database. A distributed homogenous database system contains a number of distributed databases that use similar database management systems software and identical operating systems at each site. In a heterogeneous distributed database system, the distributed databases are dissimilar in both structure and architecture. It is possible for a node to have an object-oriented database while another one can have a relational database.

E. Transaction

A transaction can be defined as a group of tasks. A single task is the minimum processing unit, which cannot be divided further [8].

A transaction is initiated explicitly by issuing a begin transaction statement and terminated by the end transaction statement. All the statements which are executed between begin and end transaction statement are what constitutes a transaction.

The collection of tasks to be executed must appear as a single unit to a user. Since a transaction is indivisible, either all the tasks execute successfully to completion or non-succeed. This property is called atomicity. If a transaction begins to execute but fails for whatever reason, any changes to the database that the transaction may have made must be reversed.

The database transaction should respect the four properties stated called atomicity, consistency, isolation, and durability (ACID).

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 7.122|

||Volume 10, Issue 1, January 2021||

DOI:10.15662/IJAREEIE.2021.1001002

F. Atomicity in Distributed HeterogeneousDatabase

There are two methods that are used to implement atomicity in a distributed database, which are index and value, and the use of private workspace. These two methods ensure that the original index and value can be found before the transaction is committed and changes are written are to the database. By using a commit method called the 2-Phase Commit, there is a guarantee that all the changes to the database are atomic, which will result in either a commit or abort [9].

The two-phase commit protocol assumes that for every distributed transaction, there exists at least one coordinator which is responsible for arranging activities and synchronizing between distributed servers. The rest of the nodes are referred to as slave nodes.
The protocol consists of two phases:

Phase 1: Prepare Phase

- After each participating node has completed its transaction, it sends a “DONE” message to the transaction coordinator. Once the coordinator has received these messages from all the participating nodes, it sends a PREPARE message.
- Each slave responds to the “PREPARE” message by sending a “READY” message back to the coordinator.
- If any of the slaves respond with a ‘NOT READY’ message or does not respond at all, then the coordinator will send a global “ABORT’ message to all the other slaves.
- Only after receiving an acknowledgment from all the slaves that the transaction has been aborted does the coordinator consider the entire transaction aborted.

Phase 2: Commit phase

- If all the participating nodes send a “READY” message to the transaction coordinator, it then issues a “COMMIT’ message to all the participating nodes for that particular transaction.
- Each slave node executes the Commit and returns a “DONE” acknowledgment message back to the coordinator.
- Only after receiving the DONE messages from all the slaves will the coordinator consider the entire transaction as completed.

Blocking Problem in 2PC

In the 2PC protocol, the failure of the coordinator will lead to the Global-Commit or Global-Abort messages not being delivered to the participants. If such a situation happens, the participants are blocked until the coordinator becomes available to send the termination decision.

Abuya et al., 2018 proposed an algorithm that uses a backup coordinator to overcome the blocking problem caused due to the unavailability of the coordinator. A primary coordinator generates a prepare message for a 2PC distributed transaction, including an address of a backup coordinator. The primary coordinator maintains a transaction log of the distributed transaction, wherein the transaction log is accessible to both the primary coordinator and the backup coordinator. The prepare message is sent to multiple participants. The primary coordinator fails over to the backup coordinator without interrupting the distributed transaction.

IV. METHODOLOGY

A. Structured Interviews

Structured interviews were conducted with some industry experts in the Zambian banking sector to provide an insight into the problem of transaction reversal for failed ATM transactions. The reason for using this method was to get an in-depth understand of the ATM reversal problem.

B. Process Mapping

A list of ISO 8583 transaction messages and operations were identified. A new mapping was then created to ensure that they conform to the definition of the two-phase commit protocol with the following assumptions.

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 7.122|

||Volume 10, Issue 1, January 2021||

DOI:10.15662/IJAREEIE.2021.1001002

- The national financial switch acted as the transaction coordinator due to its central location and ability to provide a linkage between different nodes.
- The Issuer and the Acquirer were the nodes or slaves but still retained their transaction execution autonomy just like in the two-phase commit.
- A transaction is always initiated through the acquirer, which passes over control to the coordinator.

Various points of failure were then simulated within the process map, and the recovery actions from the issuer, the acquirer, and the coordinator were observed according to the two-phase protocol

C. Simulation

The research proposed a transaction state incrementing algorithm, which was then simulated. The operating principles of this algorithm were defined as below:

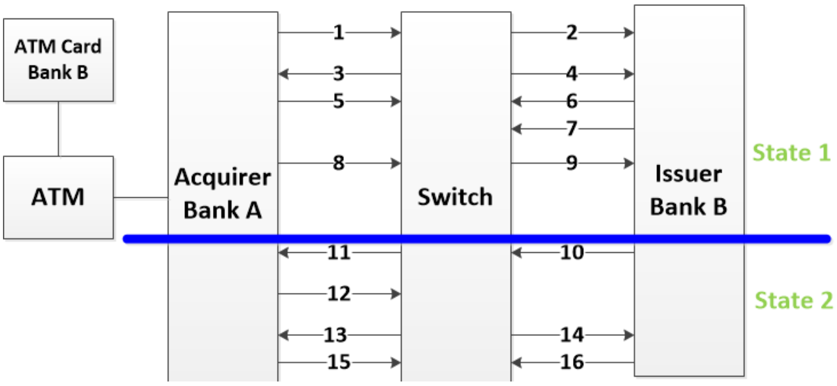

Figure 2: Transaction State Incrementing Algorithm

- 1 A transaction always starts by the acquire sending a transaction initiation request to the payment network.
- 2 Payment network sends a connection request to the issuer
- 3 & 4 once connected coordinator(Switch) starts the transaction at the acquirer and issuer
- 5 & 6 both the acquirer and the issuer acknowledge and change their transaction state to 1
- 7, transaction authorization ISO message is sent from the acquirer to the switch, which routes it to the issuer in 8
- 9, the issuer authorizes the transaction by debiting the account and immediately increments its transaction state to 2
- The payment network sends the authorization request to the acquirer in 10. Acquirer processes the request by dispensing the money through the ATM and immediately changes its transaction state to 2.
- The acquirer sends a transaction completed message in 11
- Switch (coordinator) requests transaction to commit at acquirer and issuer in 12 and 13
- Acquirer and issuer terminate the transaction and respond in 14 and 15.

Algorithm Rules

- A node, which is transaction state 1 can terminate a transaction on its end and reverse any changes done if an error is encountered.
- A node that is in transaction state 2 cannot terminate a transaction if an error occurs, but however, it cannot reverse any changes which were made by the transaction without waiting for instructions from the coordinator.
- Termination of a transaction results in the release of any held resources by that transaction.
- At any given time t, the transaction state of the issuer S(I) will always be greater or equal to the transaction state of acquirer S(A) t $S(I) \geq S(A)$. This definition will protect the system from a situation where the money is dispensed by the ATM, but a message of this successful transaction is not delivered to the coordinator hence resulting in a

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 7.122|

||Volume 10, Issue 1, January 2021||

DOI:10.15662/IJAREEIE.2021.1001002

reversal. In this case, the algorithm guarantees that such a reversal does not occur until the state of the acquirer is determined.

Simulation Tools

Neapay, a payment simulation software that can simulate ISO 8583 messages, which are used in the ATM transaction exchange, was used. The application is developed in Java and allows business logic to be defined using JavaScript. Three nodes, the acquirer, issuer, and payment switch, were created. Business logic was created according to the algorithm.

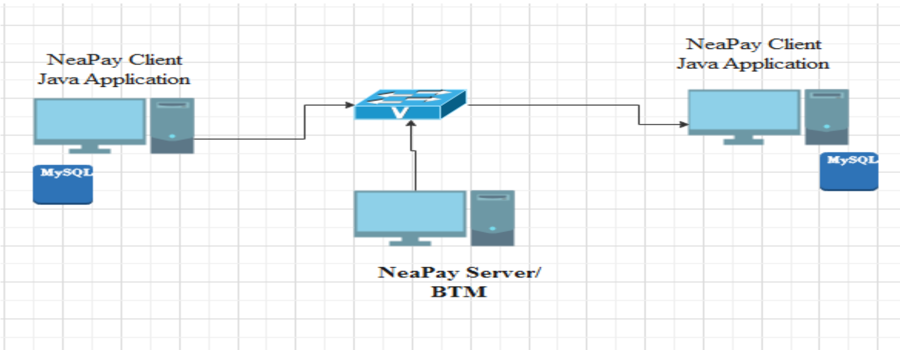

Figure 3: Network Setup

Automatically Induced failure

In some testing phases, failure was automatically induced by intentionally inserting JavaScript timeouts during the execution cycle.

Total transaction time T was defined where T = Σ (to + t2 + t3). Where t denoted the time it took to send a system message to the receiving entity and carry out a specific operation O. For this simulation, only this time was set to 4 seconds.

Failure Model Algorithm

Atomicity ensures that all operations are treated as a single logical unit in which all the sub-operations succeed or none succeed at all.

$T = \{O^1, O^2, O^3, O^4, O^5, \ldots.., O^n\}$ where T is a set denoting a series of operations O executed in an incremental order as a single logical unit in order to successfully carryout a transaction.

S^n denote a set of operations

n denote the simulation sequence number where n > 1

$S^n \subseteq T$.

Because the set of operations were designed to execute in a logical order such as $O^{n,}$ followed by O^{n+1}. Simulation S^n, therefore, consisted of a set of operations $S^n = \{O^1 \ldots O^n\}$. Failure was induced at operation O^{n+1}.

Operations were defined within the transaction operating framework as shown below for an ATM withdrawal.

IV. RESULTS AND DISCUSSION

A. Semi-Structured Interviews

During the semi-structured interviews, it was established that the assumption by the researcher that all failed transactions when conducting an interbank ATM transaction cannot be reversed is false. It was discovered that some ATMs are actually able to reverse these transactions.

It was also established the functionality to reverse transactions is available to all ATMs as specified in the ISO 8583 transaction reversal messaging standard, but this functionality has been disabled on some older model ATMs to protect from transaction reversal fraud. These older models are unable to effectively detect and prevent this type of attack. The majority of ATM terminals in the country are still using these models, and it would require significant capital investments by the banks to upgrade these machines.

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 7.122|

||Volume 10, Issue 1, January 2021||

DOI:10.15662/IJAREEIE.2021.1001002

B. Process Mapping

The process mapping proved that the two-phase commit protocol in its current definition if used to manage transaction atomicity in ATMs, would cause the following problems:

1. Blocking

If a failure is encountered in the payment network (coordinator), the customer's account balance will remain in a locked state until an instruction to commit or abort is issued.

2. Transaction Reversal Fraud

According to the protocol, any errors encountered during the execution will result in a global abort of the entire transaction. If a communication failure occurs just after the dispensing of cash, the debit that was done on the customer's bank balance would be reversed since the transaction has timed out. A customer can then go and withdraw this money from another ATM. Because of this, the system would become very vulnerable to transaction reversal fraud since different events can be used to deliberately trigger a transaction failure at the ATM terminal.

C. Transaction State Incremental Algorithm

The following were the results of the simulations:

Simulation 1: Control Simulation

This simulation was successfully executed to completion, and the transaction states were incremented according to the algorithm.

Simulation 2: Authorization Failure

A transaction authorization failure was simulated by deliberately causing the transaction to timeout at the Acquirer. Because no authorization had been done yet, the transaction state of both the Issuer and the Acquirer was 1. The coordinator issued a rollback transaction instruction to the Acquirer but could not reach the issue. The Issuer aborted this transaction due to timeout and rolled back any changes made since it was in state 1.

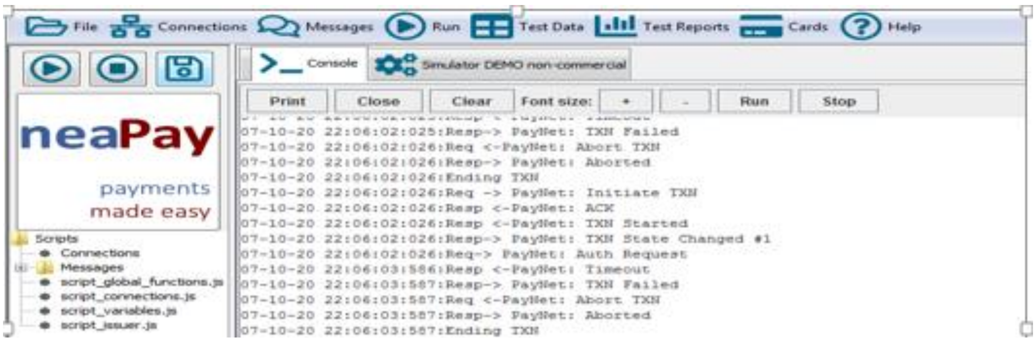

Figure 4: Recovery at Issuer

Simulation 3: Failure after Authorization

A failure at the Acquirer was simulated just after the authorization of the transaction but before the dispensing of cash. In this case the transaction state of the Issuer was 2 while that of the Acquirer was 1. The transaction was terminated at both the Issuer and the Acquirer but no rollback was executed at the Issuer because it was in state 2. When the Acquirer became available it communicated its state to the coordinator which then issued a roll back of any changes that were made at the Issuer. For that period that the Acquirer was unavailable to communicate its transaction status, the money

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 7.122|

||Volume 10, Issue 1, January 2021||

DOI:10.15662/IJAREEIE.2021.1001002

was not available for the customer.

Simulation 4: Payment Network (Coordinator) Failure

In Simulation number 4, failure was induced at the payment network (coordinator) by making it unavailable. Because the transaction state of the issuer was 2 the transaction did not automatically reverse after the timeout but just aborted the transaction. When the coordinator become available it responded to the issuer to reverse the transaction which was successfully concluded.

V. CONCLUSION

The research established that the error of a bank account being debited but cash not dispensed, which occurs when carrying an inter-bank ATM cash withdrawal, cannot be reversed automatically in Zambia because this functionality has been disabled to protect the system from transaction reversal fraud. Legacy ATM models, which are mostly used in Zambia, do not have the sophistication to detect and prevent this type of attack. However, it also established that some newer ATM models could automatically reverse a failed transaction by crediting the affected customer's bank account since there are more capable of preventing transaction reversal fraud. Upgrading all the legacy ATM models would require a significant capital expenditure by the banks.

The research also established that the current definition of the two-phase commit protocol, even though it can be effectively used to maintain transaction atomicity in distributed heterogeneous database systems, it, however, cannot be used in ATM transactions because of the blocking problem and its vulnerability to transaction reversal fraud.

A transaction state increment algorithm can successfully be used to implement the atomic execution of the transaction. Even the algorithm does guarantee real-time atomicity. It does, however, eliminate the need for manual transaction reversal.

REFERENCES

1. Luchembe, C. Siame “Mobile Money for increased financial access”, May 2018.
2. [Online]. Available: https://www.fsdzambia.org/mobile-money-for-increased-financial-access-weve-got-your-back/. [Accessed Mar. 18, 2020].
3. M. Rouse "Interoperability",[Online]. Available: https://searchapparchitecture.techtarget.com/definition/interoperability. [Accessed Apr. 11, 2020].
4. Allen, D. K., Karanasios, S., & Norman, A. (2013). Information sharing and interoperability: the case of major incident management. European Journal of Information Systems, 10.1057/ejis.2013.8.
5. Clark, G. Camner,2014. A2A Making Mobile Money Schemes Interoperate, Consult Hyperion V. Sundararajan, “What is High Performance Computing?” Scientific and Engineering Computing Group: Centre for Development of Advanced Computing (CDCC), [online]. Available: http://www.iacs.res.in/What_is_HPC.pdf. [Accessed Oct. 5. 2015].
6. B.Ng’andu 2018. National Financial Switch. Bank Of Zambia, 2018
7. Dally, W.J. and Towles, B.P., 2004. Principles and practices of interconnection networks. Elsevier
8. Distributed DBMS Concepts, [Online]. Available: https://www.tutorialspoint.com/distributed_dbms/distributed_dbms_quick_guide.htm. [Accessed Apr. 08, 2020].
9. Gligor, 1985. Transaction Management In Distributed Heterogeneous Database Management Systems, Journal Inform Systems Vol. 11, 1986
10. Tanenbaum, van Steen, 2002: Distributed Systems, Principles and Paradigms; Prentice Hall
11. BTech Group “Interoperability and Digitization”, [Online]. Available: https://btech-group.com/service/brandguid-line/. [Accessed Apr. 08, 2020].
12. P. Mutimushi “The 2019 Annual Report”, Zambia Information and Communications Technology Authority, 2019.
13. T. Sivakumar, G. Askok, k. Venuprathap, 2013. Design and Implementation of Security Based ATM theft Monitoring system. In International Journal of Engineering Inventions Vol 3, Issue 1 (August 2013)
14. Goyer, P., P. Momtahan, and B. Selic. "A Fault-Tolerant Strategy for Hierarchical Control in Distributed Computer Systems," Proc. 20th IEEE Symp. on Fault-Tolerant Computing Systems (FTCS20), (IEEE CS Press, 1990), pp. 290-297
15. Tanenbaum, van Steen, 2002: Distributed Systems, Principles and Paradigms; Prentice Hall.

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 7.122|

||Volume 10, Issue 1, January 2021||

DOI:10.15662/IJAREEIE.2021.1001002

16. M. Biallas, A. Fook, R. Armelle "IFC Mobile Money Scoping: Country Report Zambia", Internation Finance Corporation, World Bank Group.
17. Chad Ferguson, Sandra Brey, SAS Institute Inc., Cary, NC. Developing Data-Driven Applications Using JDBC and Java Servlet/JSP Technologies
18. N. Kumar, L. Sahoo, and A. Kumar, "Design and implementation of Three Phase Commit Protocol (3PC) algorithm," 2014 International Conference on Reliability Optimization and Information Technology (ICROIT), pp. 116–120, 2014
19. H.J. Salah, R.Rimiru, M, Kimwele. An Alternative Model to Overcoming Two Phase Commit Blocking Problem, International Journal of Computer (IJC) (2017) Volume 26, No 1, pp 71-88
20. Gavriluu, A. Gavriluu,M. Agop,2019. Extended Minimal Atomicity through Nondifferentiability:A Mathematical-Physical Approach, Advances in
21. I. Wojciechowska "The basics of ACH credit ", [Online]. Available: https://fin.plaid.com/articles/ach-credit-transaction/. [Accessed Mar. 16, 2020].
22. . McAndrews "The automated clearinghouse system: moving toward electronic payment", in Business Review Federal Reserve Bank of Philadelphia, Aug, 1994
23. Electronic Project Focus "How ATMs Work", [Online]. Available: https://www.elprocus.com/automatic-teller-machine-types-working-advantages/. [Accessed Mar. 17, 2020].
24. S. Kumar "Introduction to ISO 8583", [Online]. Available: https://www.codeproject.com/Articles/100084/Introduction-to-ISO. [Accessed Mar. 17, 2020].
25. Zambia Invest "World Bank Encourages Telecom Sector To Grow Zambia Mobile Money System", [Online]. Available: http://www.zambiainvest.com/telecoms/world-bank-encourages-telecom-sector-to-grow-zambia-mobile-money-system. [Accessed Mar. 17, 2020].