Vol 8. No. 4, December, 2015
ISSN 2006-1781

African Journal of Computing & ICT

Light-Weight Client/Server Transaction Processing Architecture for Ubiquitous Computing Environment

P. Spencer
Department of Computer Science
Ignatius Ajuru University of Education
Rumuolumeni, Rivers State, Nigeria.
patsyspency2013@hotmail.co.uk

E.O. Nwachukwu
Department of Computer Science
University of Port Harcourt
Rivers State, Nigeria
enoch.nwachukwu@uniport.edu.ng

ABSTRACT

Database transaction processing systems take a central position in all information-based community. As such, the correctness and efficiency of database management systems in ubiquitous computing environment is essential. The optimal performance of information processing system in ubiquitous computing environment is greatly influenced by the structural organisation of system components and communication technique of the system. Existing transaction processing models suffer an unbearable setback which may have been due to poor structural organisation or communication technique employed. For optimal realisation of ubiquitous computing, the structural organisation of system components and communication techniques must provide effective support for real-time processing. This work therefore reviews existing system architectures with the aim of identifying the most suitable system architecture for ubiquitous computing environment. The choice of a three-tiered, light-weight client/server system architecture is to provide effective load management architecture and to avoid the negative effect of short battery life and other issues associated with mobile user devices on transaction processing in ubiquitous computing environment.

Keywords: Ubiquitous Computing, Transaction, Client, Server, Architecture, and light-weight

African Journal of Computing & ICT Reference Format:
P. Spencr & M. E.O. Nwachukwu& (2015): Light-Weight Client/Server Transaction Processing Architecture for Ubiquitous Computing Environment. Vol 8, No. 3. Pp 201-.208.

1. INTRODUCTION

Generally, an architectural design defines a system's pattern structural organisation [1]. It determines the components (That is, objects and methods in object-oriented design approach) and connectors (that is, communication infrastructure) that can be used in the design together with a set of limitations (relating to topology, structural description, and execution semantics) on how they can be put together. The optimal performance of any information processing system is greatly influenced by the system's architectural design and employed communication technique. Transaction processing Systems in ubiquitous computing environment are not left out of this phenomenon. However, the performance of existing transaction processing systems for ubiquitous computing may not have satisfactorily exhibited the full potentials of real-time processing, greatly due to poor architectural design of the information processing system.

Focusing on the structural organisation of transaction processing components in ubiquitous computing environment, this work first takes a look at existing transaction processing architectures and then presents a three-tier, light-weight client/server architecture as the most suitable system architecture for ubiquitous computing environment. With this, the negative effect of short battery life and other unhealthy interferences associated with mobile user devices are taken care of in addition to effective system load distribution.
.

2. MATERIALS AND METHODS

Employing an object-oriented methodology, the components of the proposed system architecture (mobile devices, Application Server, Transaction server, and Processing Units (data sources)) are seen as different objects coming together to achieve a common goal. They are treated as objects because they preserve the integrity of their individual representations in a transparent manner and communicate with each other via

Vol 8. No. 4, December, 2015
ISSN 2006-1781

appropriate communication protocol and information management support systems [2]. This strategic organisation provides effective load management architecture.

To appreciate the significance of the proposed architecture on ubiquitous computing, an overview of the evolution of transaction processing architectures and their impact on ubiquitous computing is first presented. After which a critical analysis of the proposed architecture and its impact on ubiquitous computing is emphasised on. Ubiquitous computing paradigm is concerned with the ability of a user with a mobile computing device (wearable and handheld) to be able to access information residing in different computers as though the information is in the user's computer [3]. Information processing in ubiquitous computing environment with inherent distributed database systems could be very challenging [4]. Especially, distributed systems like security and error handling, are inherently difficult [4]. Software applications developed to support ubiquitous computing are faced with challenges that come with the technology. The challenges are mainly related to mobility, interconnectivity, and context-awareness. These challenges affect the smooth flow of transaction processing activates.

A transaction is a collection of several operations that form a single logical unit of work [4]. However, a user (That is, the physical user) sees a transaction as a single operation A review of the evolutional trend of transaction processing system architectural designs shows a shift from centralized single-user and multi-user architecture to a variations of client/server transaction processing architectures.

Let's look at these architectural design trends and their impact on ubiquitous computing. Having in mind that they all have the following basic transaction processing components:

i. End-user device
ii. Front-end program
iii. Request controller
iv. Transaction server
v. Database system

2.1 Centralized single-user and multi-user architecture
Figure 2.1 represents a centralized single-user transaction processing architecture whereas Figure 2.2 represents a centralized multi-user transaction processing architecture. In the centralized architecture, the user module and database management system (including data sources) reside in the mobile device [5]. Every transaction management system is expected to ensure that the ACID (Atomic, Consistency, Isolation, and Durability) properties of a database are upheld. This means that:

1. a transaction happens in its entirety or not at all should there be any kind of failure.
2. If the database is consistent before an execution, even after the execution of a transaction, the database remains consistent. This property is also known as correction requirement.
3. Multiple transactions execute concurrently (i.e. request CPU attention at the same time) occur transparently without conflict.
4. Transaction management guarantees the successful completion of a transaction and ensuring that all updates carried out on the database persist (that is, remains the same) even if there is a system failure after the transaction completes execution.

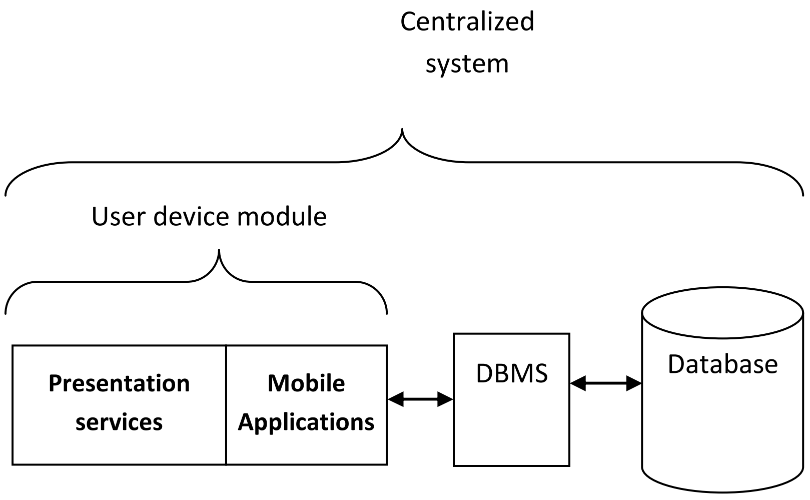

Figure 2.1: Centralized Single user transaction processing system architecture

Vol 8. No. 4, December, 2015
ISSN 2006-1781
African Journal of Computing & ICT

The centralized single user architecture of a transaction processing system is made up of the user module that deals with the management of presentation objects such as forms and procedures for information going to/from the user interface [5]. It also has the responsibility of managing user request (that is, transaction) and interaction directly with the database management system (DBMS). This phenomenon creates no attention to the ACID properties of database [5].

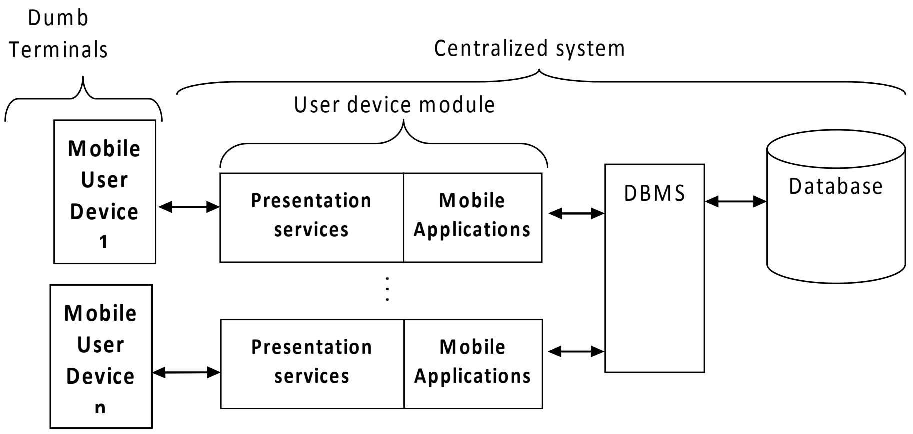

Figure 2.2: Multi- User transaction processing system architecture

The centralized multi-user architecture has a system organization where dumb terminals are connected. The dumb terminals are connected via a communication link to a mainframe computer. Unlike the centralized single-user architecture, in the centralized multi-user system architecture, the application and presentation services are handled in the mainframe (which serves as a server) allowing more than one user connection [5]. Since different user modules are given access to the same database items, ACID property check is required. The database management system takes responsibility of managing transactions ensuring that Atomicity, Isolation (possibly via interleaved schedule), and Durability are properly handled in order to maintain data integrity. Transaction processing system architectural design moved from the centralized (non-layered) architecture to the client/server (layered-layered) architecture [1] and allows the organisation of system components to be done in two ways. One way is by dividing the structure into two tiers and the other is dividing the structure into three tiers. [6] explains that one of the demands of database system transaction management is to achieve a high degree of concurrency by taking into consideration the semantics of high-level operations and that implementing of such operations must give attention to conflict on the storage representation levels. In order to achieve these characteristics, layered system architecture is required. The following sub-sections present two variations of layered transaction processing system architectures (Figures 2.3 and 2.4) and the proposed system architecture for ubiquitous computing which is based on the three-tiered layered model.

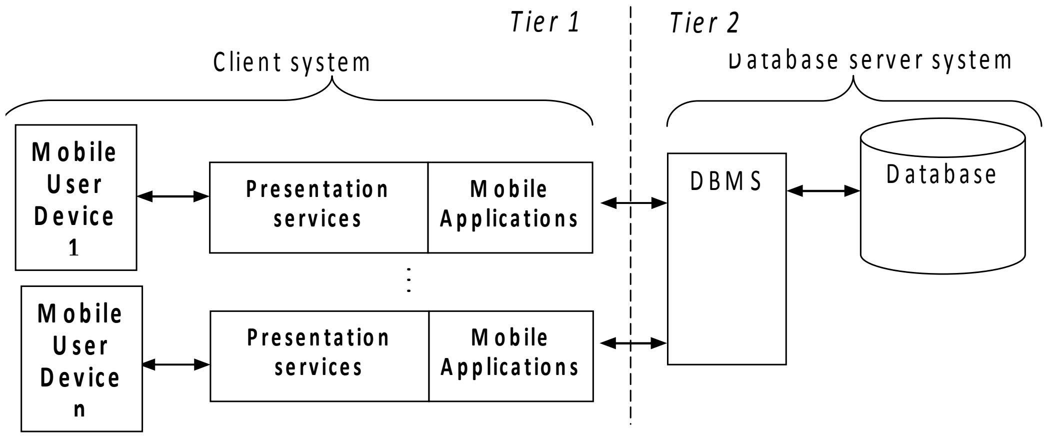

Figure 2.3: Two-tier client/Server transaction processing system architecture

Vol 8. No. 4, December, 2015
ISSN 2006-1781

2.2 Two-tier Client Server architecture

Figure 2.3 represents two-tier client/server (Service user/service provider) [5] architecture of a transaction processing system. In this model, the stored procedure interfaces, presentation services and application services are located in the mobile user device found in the 1st tier while stored procedures are kept and maintained at the server site found in the 2nd tier. This phenomenon makes use of fewer number of communication links and reduces client related interference on the server when compared with the centralized architecture.

2.3 Three-tier Client Ser architecture

Figures 2.4a and 2.4b represent the varieties of three-tiered architecture.

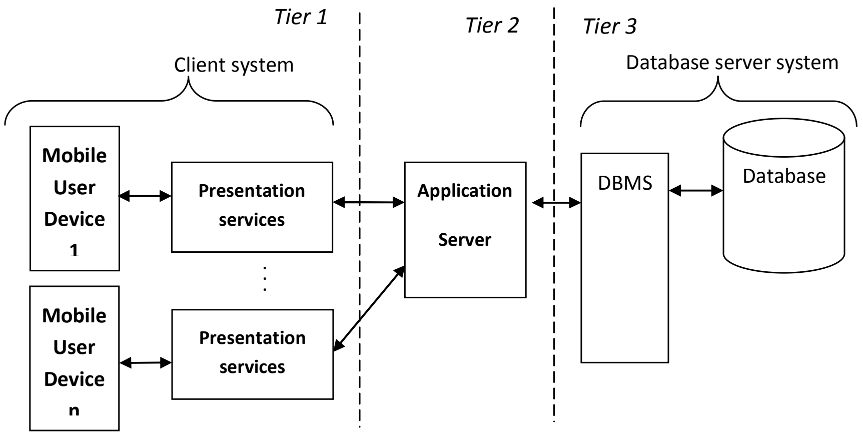

Figure 2.4a: Three- tiered architecture of a transaction processing system showing the separation of the application server from the client machine.

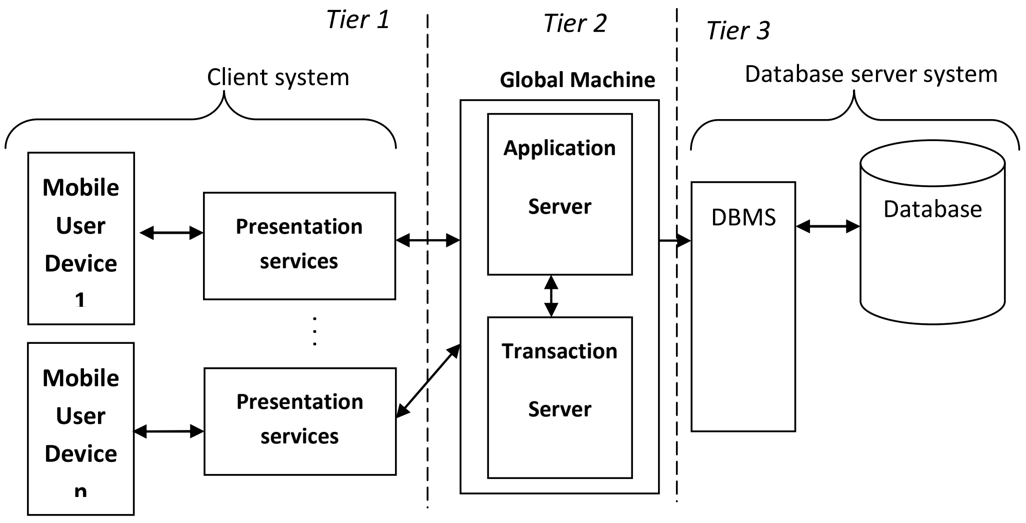

Figure 2.4b: Three- tiered architecture of a transaction processing system showing the application server and the transaction server forming the middle tier.

Figure 2.4a is an illustration of a three-tier architecture where the application module is separated from the presentation module. So the client machine now holds the user interface and the presentation services only. The application services are now found in the application server machine. In this way, different client machines can communicate with the database server through the application server.

Responsibilities of the Application Server are:

i. Set transaction boundaries.
ii. Implement user request as a sequence of tasks (that is, doing the functions of a controller).
iii. Act as a router as it affects management of distributed transactions and load balancing.
iv. Manage clients' requests by applying multi-threading skills.

Figure 2.4b is an illustration of a three-tier architecture where the DBMS is relieved of matters concerning stored programs. The stored procedures are moved from the database server to a separate server known as Transaction Server.

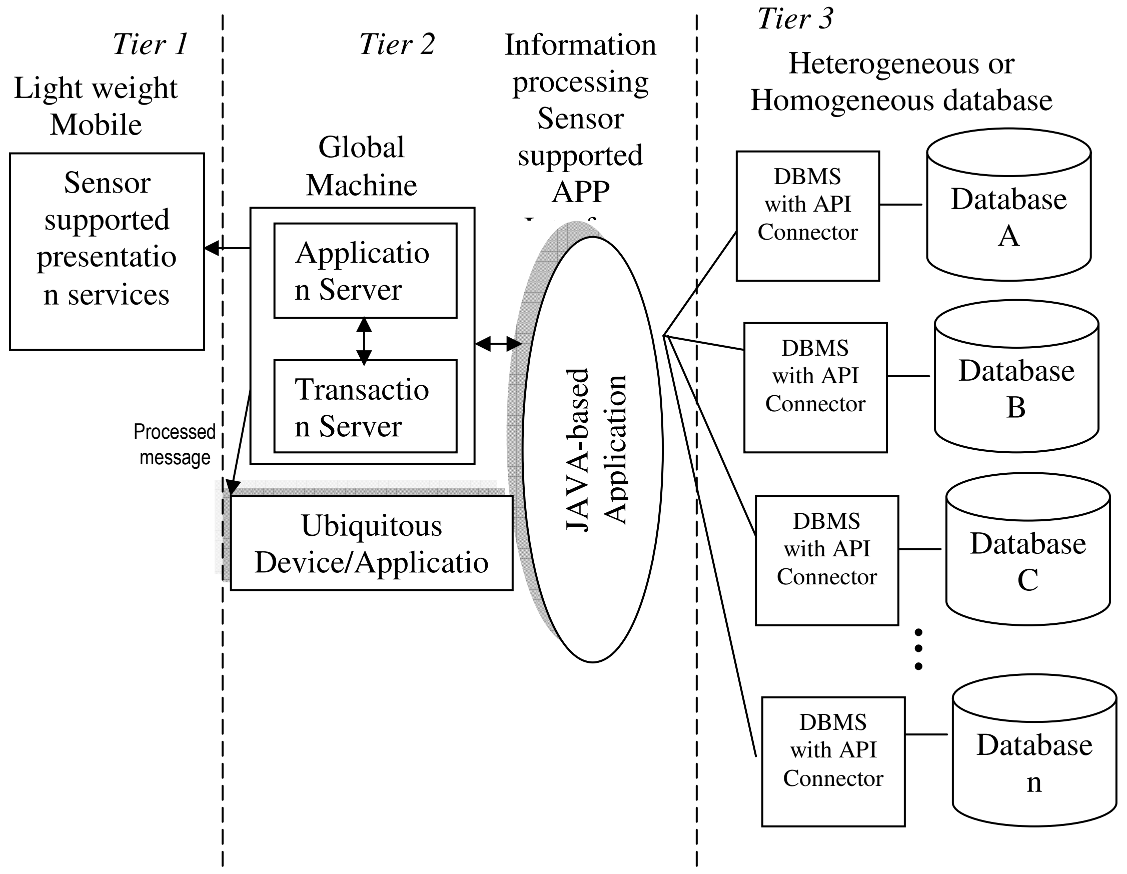

Figure 2.5: A block diagram illustrating the proposed transaction processing architecture for ubiquitous computing environment.

Vol 8. No. 4, December, 2015
ISSN 2006-1781

The transaction server is directly connected to the database server. The sole function of the transaction server is to manage transaction segments. The application server uses the transaction server to execute stored procedures.

2.4.Proposed Transaction Processing Architecture for ubiquitous computing

Haven examined the advancement in architectural designs for transaction processing systems, this study proposes the deployment of a three-tier architecture in ubiquitous computing environment. This is because a transaction server dedicated for the management of transaction execution is introduced. In this way, the DNMS is now relieved of tasks relating to the functionalities of stored programs. The transaction server runs on top of the DBMS. Figure 2.5 is a block diagram illustrating the proposed transaction processing architecture for ubiquitous computing environmen

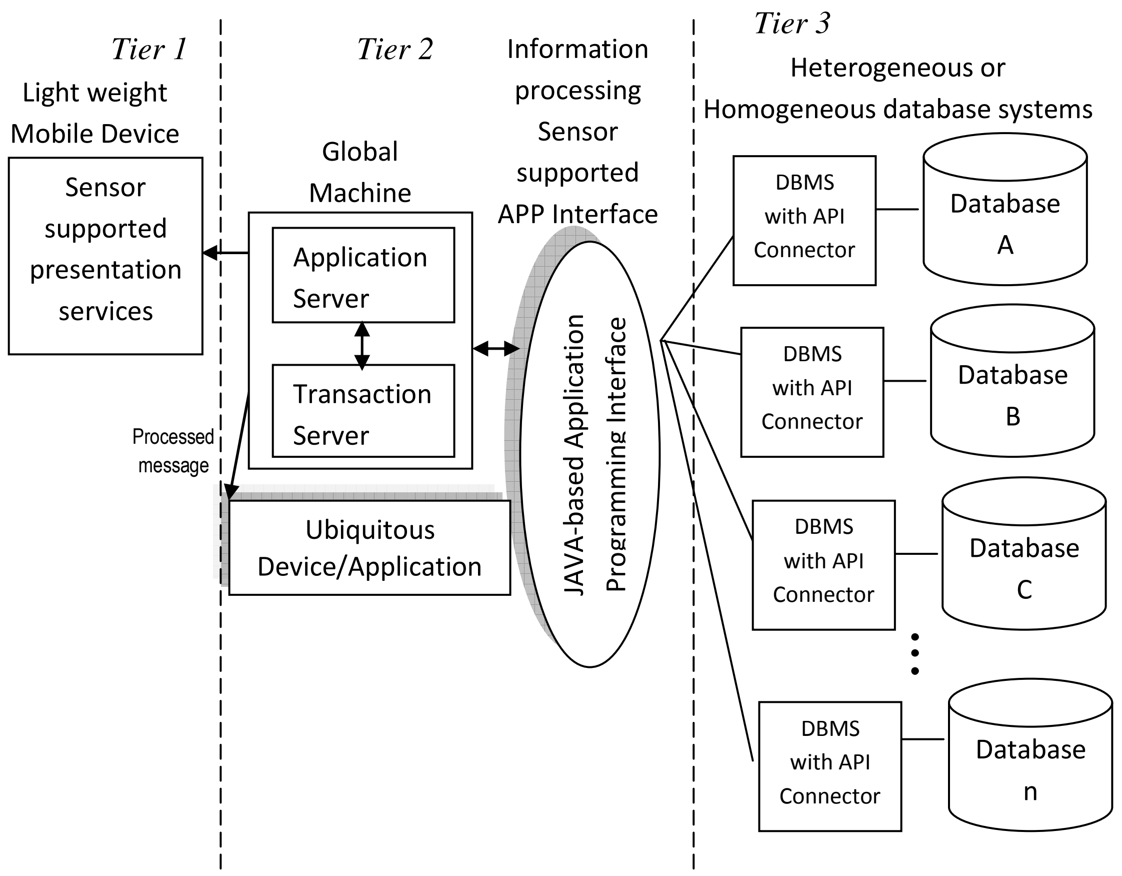
Tier 1
Light weight Mobile Device
Sensor supported presentation services
Processed message
Tier 2
Global Machine
Application Server
Transaction Server
Ubiquitous Device/Application
Information processing Sensor supported APP Interface
JAVA-based Application Programming Interface
Tier 3
Heterogeneous or Homogeneous database systems
DBMS with API Connector
Database A
Database B
Database C
Database n

Figure 2.5: A block diagram illustrating the proposed transaction processing architecture for ubiquitous computing environment.

Vol 8. No. 4, December, 2015
ISSN 2006-1781
African Journal of Computing & ICT

In the first tier of figure 2.5, mobile devices with inbuilt information management sensor are located using an agent-based message passing technology, and effective logging system, user preferences are transmitted to the application server **saddled with the responsibility of taking log of the status of users requests profiles and the transmitting requests to the** transaction server.

The Application server and transaction server are located in the second tier. Removing the application server from the mobile device makes it possible for unannounced disconnection of the mobile device not to have effect on the execution of users' request. This in turn reduces processing latency. Feedbacks from the transaction server are logged in the application server and transmitted to the mobile device once the mobile device reconnects to the system.

The transaction server saddled with the task of managing the actual execution of user requests (that, transactions) connects to the database system via information processing sensor supported application interface. The transaction server on getting a user request from the application server broadcast request for readiness of database systems for a job to available data sources. Once that is established, the transaction server keeps track of the progress of active transaction until a logical conclusion is reached. The data sources and database management systems are located in the third tier of the system.

3. RESULT AND DISCUSSION

This study creates an effective load management mechanism where:

The mobile user device is made a light weight processing device saddled with the task of:

i. collecting location-based and non location-based user preferences in complex environment
ii. Providing user preferences to the application server
iii. Facilitate communication (via voice, text, video, etc.) with support agent system.
iv. Compute/display personalized reports, choices, and reminders
v. Compute/display user preferences
vi. Compute/display errors and breakdowns.

The Application Server (App Server) is saddled with the task of:

i. Interfacing with support agent system for mobile user request, user profile and request profile
ii. Log user and request profiles
iii. Transmit user request (transaction) and logs to the transaction server
iv. Monitor user preferences for possible online updates
v. Transmit online updates to transaction server
vi. Get feedback and associated logs from transaction server and
vii. Prompt mobile user of success or failure of execution of request.

The Transaction Servers (TS) is saddled with the task of:

i. Interfacing with the application server for new transaction and associated logs
ii. Interface with support back-end agent systems (That is. State Information management system) for the Identification of eligible processing units.
iii. Distribute transaction branches to eligible processing units that are ready to participate in the execution of the new transaction
iv. Monitor execution process for possible online updates, failure, migration and successful completion.
v. Abort and terminate failed transactions and communicate transactions' states to the application server.
vi. Commit successful transactions and communicate result and transactions' states to the application server.

The Database System/Servers are saddled with the task of:

i. Telemetry data (all database related activities) and
ii. Sharing itineraries with mutual understanding

The introduction of the middle tier and taking away the core telemetry functions from the mobile user device obviously takes care of the problems of increased message overhead and latency associated with the two-tier client/server model. Making the transaction server proactive (through agent technology) in managing database transactions further enhances the performance of the proposed architecture in terms of giving effective support to location dependent transactions and avoidance of possible failure of active transactions that may be caused by network failure or system breakdowns [7]..

4. CONTRIBUTIONS TO KNOWLEDGE

The importance of database management in all information-based systems can never be over emphasised. So the correctness and efficiency of database transaction management systems especially in ubiquitous computing environment is essential. This paper gives a clear understanding of how structural organisation of components of transaction processing systems impact on the performance of transaction processing systems in ubiquitous computing environment. After a critical analysis of the operational principles of existing centralised, two-tier client/server and three-tier client/server system architectures on which ubiquitous computing is deployed, this paper identified poor load balancing and unnecessary data communication latency as major setbacks associated with the centralised and two-tier client/server system architectures. The paper then proposes a light-weight (as it affects responsibility) client on a three-tier client/server system architecture. Locating just the client

Vol 8. No. 4, December, 2015
ISSN 2006-1781
African Journal of Computing & ICT

presentation programmes in the first-tier; the client application programmes (server) and the database server in the second-tier; and then the database system in the third-tier, have shown that the proposed three-tier system architecture effectively manages system load and data transfer latency better than the centralised and two-tier system architecture.

5. CONCLUSION

This work presents an overview of transaction processing system architectures then presents the three-tier transaction processing architecture as the most befitting system architecture for ubiquitous computing environment where the mobile user device is located in the 1st tier, the Application and Transaction Server are located in the 2nd tier and the database system and associated tools are located in the 3rd tier. The proposed three-tier transaction processing architectural design for ubiquitous computing environment is shown to effectively support ubiquitous computing as it supports effective management of system load which in turn takes care of the problems of increased message overhead and latency associated with the two-tier client/server model.

6. RECOMMENDATION

The proposed system architecture for ubiquitous computing is recommended for researchers' consideration. It is also recommended for deployment in information-based support system by software developers.

7. FUTURE WORK

Future study will attempt to look for the appropriate communication technique to implement on the proposed three-tier structural
design for ubiquitous computing environment

Acknowledgement
Thanks to Enoch O. Nwachukwu, Japhet R. Bunakiye, and Emmanuel Nyoho for their technical support and helpful discussions.

REFERENCES

[1] Garlan, D., & Shaw, M. (1993). Introduction to Software Architecture. Ambriola, V., & Tortora, G. (Eds.). Advances in
[2] Software Engineering and Knowledge Engineering (vol. 2). World Scientific, Singapore.

[3] [Filip, M. J., Karunungan, K. L., Kramer, J. C., Lee, L. C., Moore, D. L., Shih, C. C., & Sydir, J. J. (1995). U.S. Patent No.
[4] 5,414,812. Washington, DC: U.S. Patent and Trademark Office.

[5] Nwachukwu, E. O. (2010). Information Technology: The Albatross of Our Time:An Inaugural Lecture. University of Port Harcourt.

[6] Puder, A., Römer, K., & Pilhofer, F. (2006). Distributed systems architecture: A Middleware Aproach. Elsevier, UK

[7] Gray, J. and Reuter, A., (1992). The architecture of Transaction Processing Systems. In Transaction Processing
[8] Concepts and techniques chapter 23. Burlington, Massachusetts, US

[9] ilberschatz, A., Korth, H. F., & Sudarshan, S. (2006) Database System Concepts McGraw Hill, NewYork.

[10] Weikum, G. (1991). Principles and Realizationstrategies Of Multilevel Transaction Management. ACM Transactions on Database Systems (TODS), 16(1), 132-180.

[11] Chong, C. Y., & Kumar, S. P. (2003). Sensor Networks: Evolution, Opportunities, and Challenges. Proceedings of the IEEE, 91(8), 1247-1256.