Process and Database Modelling of a University Bursary System: A Perspective of Cash Office

Friday Yakubu[1], Barroon Isma'eel Ahmad[2], Olugbemi Morenike Omowumi[3], Maria Abur Mngohol[4]
[1]Iya Abubakar Computer Center, Ahmadu Bello University
Zaria, Nigeria

[2]Mathematics Department, Ahmadu Bello University
Zaria, Nigeria

[3]Iya Abubakar Computer Center, Ahmadu Bello University
Zaria, Nigeria

[4]Iya Abubakar Computer Center, Ahmadu Bello University
Zaria, Nigeria

Abstract - It is generally recognized that System Development Method widely known as System Development Life Cycle (SDLC) is vital instrument that aids toward having an effective and efficient Information system. A lot of IT projects these days are outright failures; IT projects are cancelled before completion. Many failed systems were abandoned because the analysts tried to build a wonderful system without clearly understanding how the system would support the organization's goals, current business processes, and other information systems to provide value. Process modeling is a stage in system analysis phase that often seem to be among the least understood task in SDLC. In Nigeria, many universities are on the race of design and implementation of e-Bursary system. To have an outstanding e-Bursary System, system analysts need to understand and use the available techniques to document system requirements, and to convey the range of issues that need to be understood by system developers for developing e-Bursary Information system. This paper discusses the concepts of system process modeling, and how to develop Process model for Bursary Cash office.

Keywords: *Data flow diagram; Process Modeling, Database Modeling, Cash office System, and Bursary System.*

1. INTRODUCTION

In the Information Technology world, System Development Method widely known as System Development Life Cycle (SDLC) is vital instrument that aids toward having an effective and efficient Information system. SDLC is professionally used to produce deliverables system. SDLC as mechanism, it's a standard set of processes or steps expected to be followed on any system development project [1]. The processes may vary greatly for different projects but a common characteristic can be found that follows a problem-solving technique. The SDC approach typically has the following general problem-solving steps: System Initiation/feasibility analysis, System analysis, System design and System implementation. It is during the system analysis phase, the analyst determines the business requirements for the new system, and models the requirements.

In the SDLC, the key person is the systems analyst who analyzes the business situation, identifies opportunities for improvements, and designs an information system to implement them. Systems analysts use many graphically techniques to describe an information. One of the generally use technique is to draw a set of data flow diagrams. A data flow diagram (DFD) is a tool that uses various symbols to show how data moves through an information system but does not show program logic or processing steps [2]. A set of DFDs provides a logical model of what a system does. The aim of this paper is to broaden the understanding concepts on process modeling. The paper described at a very comprehensive level the steps, and some of the skills that are used to document Bursary Cash office requirements through the use of process model.

2. PROCESS MODELING

System models have effective place in any system development project. System analyst often constantly deals with unstructured problems during system analysis. Among methods to structure such problems is to use pictorial representation commonly known as models. A model is a representation of reality [1]. For good understanding of a system, system analyst use models to document system requirements. Process model is a formal way of representing how a business system operates [3]. It shows the processes or activities that are performed in a system and how

data moves. In order to illustrate a process model, data flow diagram is needed. DFDs reveal relationships among and between the various components in a system.

Data flow diagram symbols consist of four symbols which are processes, data flows, data stores and external entities. A process is an activity or a function that is performed for some specific business reason; A data flow is a single piece of data or a logical collection of several pieces of information; A data store is a collection of data that is stored in some way; An external entity is a person, organization, or system that is external to the system but interact with it [4]. Figure 1, shows the standard set of symbols that shall be used in this paper.

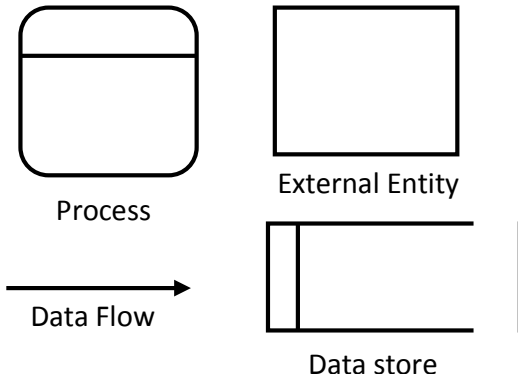

Figure 1: DFD set of symbols

When it comes to conveying how data flows through system and how that data is transformed in to the process, DFDs are the method of choice over technical descriptions for three principle reasons [4].

1. DFDs are easier to understand by technical and non-technical audiences
2. DFDs can provide a high level system overview, complete with boundaries and connections to other systems
3. DFDs can provide a detailed representation of system components.

2.1 PROCEDURES FOR DEVELOPING PROCESS MODEL

In process modeling, the highest-level of data flow diagram is known as the context diagram [1]. A context diagram is a data flow diagram of an organizational system that shows the system boundaries, external entities that interact with the system and the major information flows between the entities and the system [5]. The context diagram shows the overall business process as just one process and shows the data flows to and from external entities. The context diagram is further decomposed into the lower-level, and each process on the level can be decomposed into more explicit data flow diagram. To properly model a system using DFDs there is a set of rules [8] that must be followed:

Rules of data flow diagrams:

- At least one input or output data flow for external entity
- At least one input data flow and/or at least one output data flow for a process
- Output data flows usually have different names than input data flows for a process
- Data flows only in one direction
- Every data flow connects to at least one process

Rules of unique name in data flow diagrams:

- A unique name (verb phase), a number and a description for a process
- A unique name that is a noun and a description for a data flow
- A unique name that is a noun and a description for data store
- A unique name that is a noun and a description for external entity

Rules of Consistency and Consistency Viewpoint:

- Every set of data flow diagrams must have one context diagram.
- There is a consistency viewpoint for the entire set of DFDs.

Rules of Decomposition and Balancing

- Every process is wholly and completely described by the processes on its children DFDs.
- Every data flow, data store and external entity on a higher level DFD is shown on the lower-level DFD that decomposes it.

Rules of Data Store

- For every data store, data cannot move directly from one data store to another data store.
- Data must be moved by a process.

There are two primary types of problems associated with process modeling using DFDs: syntax errors and semantics errors. *S*yntax of the data flow diagram refers to how components are interconnected through data flows and what components constitute the subsystem being modeled, and semantics error refers to how data flows are interrelated in terms of data transformations [8]. In order to overcome these situations, system analysts must ensure that the following rules of syntax and semantics [9] are used in order to verify the correctness of the DFDs.

Syntax rules of data flow diagram:

- At least one input data flow for a process.
- At least one output data flow for a process.
- Process from external entity cannot move directly to another external entity.
- At least one input data flow for a data store.
- At least one output data flow for a data store.
- Data from one data store cannot move directly to another data store

Semantic rules of data flow diagram:

- The total number and name of external entities in context diagram are the same as in level 0 DFD.
- The total number and name of data flows between process and external entity in context diagram are same as level 0 DFD.
- The total number and name of external entities in level 0 DFD are same as context diagram.
- The total number and name of data flows between process and external entity in level 0 DFD are the same as in context diagram

3. DATA MODELLING

Systems models play an important role in systems development. Data modeling is a technique for organizing and documenting a system's data. Sometimes it is called database modeling because a data model is eventually implemented as a database.

Data modeling is a disciplined method for describing the structure and meaning of business information. The term, as used in this paper, refers primarily to entity-relationship modeling. A data model is a formal abstraction of the elements of information used or created in a particular group of business functions or activities [6]. A data model is a conceptual representation of the data structures that are required by a database. There are several notations for data modeling. Entity relationship diagram (ERD) is the actual model that is frequently used. ERD depicts data in terms of the entities and relationship described by the data.

An entity is something about which a business needs to store data [7]. In other words, entities are the persons, places, or things that are important to the organization and need to be tracked in the database. A relationship is a natural system association between one or more entities [6]. An entity has many instances. Each instance is uniquely identified by a key. A key is an attribute, or a group of attributes, that assumes a unique value for each entity instance. The relational model organizes data in tables and lets you create relationships among tables by referencing columns that are common to both—the primary and foreign keys.

4. BURSARY DEPARTMENT

Every University in Nigeria has a Bursary department with the responsibility of carrying out its Finance and Accounting functions. The Bursary Departments of the Universities are responsible for the effective coordination of all financial functions and the Bursary Department has many units including Budget and expenditure unit, Cash Office and Revenue unit, Final Accounts unit, Salaries, Loans and advances unit [10-12]. The department is generally divided into three Divisions:

1. The Bursar's Office is the heart of the bursary and is responsible for the effective coordination of all financial functions in the universities. It is comprised of the Bursar's Office, the Bursary Registry, the Bursary Secretarial Services, the Insurance Unit and the Fixed Assets Unit.
2. Treasury Services and Accounts Division is headed by a Deputy Bursar and is in charge of Cash Office, Students Accounts, Salaries, Loans, Advances and Final Accounts.
3. Finance Division also headed by a Deputy Bursar handles matters related to Revenue Mobilization, Estimates and Budgets, Expenditure Control and the Central Stores.

However, the Bursary is the most sensitive service department in the universities most especially in the present era of inadequate funding and low staff morale. The department is entrusted with the responsibilities of sourcing, care and control of funds and the rendering of accounts to the funding agencies.

In response to emerging technologies and the share inability to cope otherwise, the Bursary department at Ahmadu Bello University (ABU) has been partially computerized. Notwithstanding, there are many challenges with the implementation of IT particularly in the area of software designed to aid ABU Bursary activities. Among these challenges is getting the right system requirements before the design and implementation of the software.

To understand and get the expected system requirements, proper application of process modeling rules was observed.

5. CASH OFFICE PROCESS MODELLING

In the bursary department, cash office unit is the unit that is charge with the responsibility of m*onitoring* cash operational functions while maintaining a high level of internal controls and professional accuracy and is dedicated to providing courteous and outstanding university service, and accurate posting of accounts. The unit is responsible for the entire university's various balances of cash. This cash could be currency, personal cheques, certified cheques, money orders, traveler's cheques, or bank drafts. Cash unit functions may vary and depend on the university agreed cash management policy. However, the unit function incorporates two key responsibilities: depositing cash and recording cash received in the University's accounting records. The accurate recording of receipts involves verification and classification of the amounts. These procedures are carried out by various sections in the cash office unit and these sections are referred to as the processing sections. Figure 1 shows the Cash Office context diagram (as a process). Mails, Payment Vouchers and Cash are input of the Cash Office System while e-payment schedule, Standard Journal Vouchers Schedule (SJV) and Cash are the output of the same system.

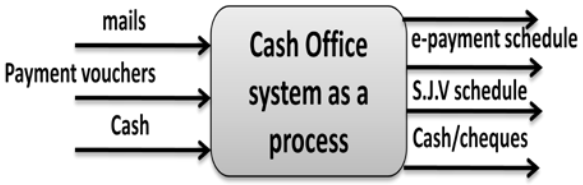

Figure 2: Cash Office Context diagram (as a process)

A. *Cash Office Process: 1st and 2nd Level decomposition*

As stated earlier, the Cash unit functions may vary and depend on the university cash management policy. However, the major functions of the Cash unit are raise schedules of payment vouchers, posting details of payment vouchers, verification and certification of payment vouchers, raising e-payment schedules, and Revenue collection. Figure 2 illustrates the context diagram. The functional data flow diagram is reflected in figure 3. Figure 3 shows the first level functional decomposition chart of the Cash Office process modeling.

Each process in the decomposition diagram is a parent process. A parent process in the chart may have two or more children processes and a child process may be the parent of its own children. In the decomposition procedures, logical process models does not include process that corresponds to forwarding or routing data which leave data unchanged such as registering incoming and outgoing mails [1].

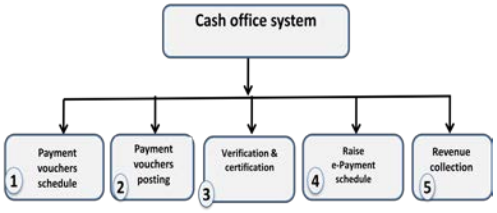

Figure 3: First level decomposition of Cash Office Context diagram.

In this case of Cash office system, each process in the first level decomposition is a parent process that has children processes. For instance, the function payment vouchers schedules has different types of schedules: SJV schedule for Local Capital Project (LCP), Overhead SJV schedule, Salary SJV schedule, and Revenue SJV schedule. Therefore, the payment voucher schedule process most be decomposed to next level of decomposition. Figure 4 illustrates the next decomposition level.

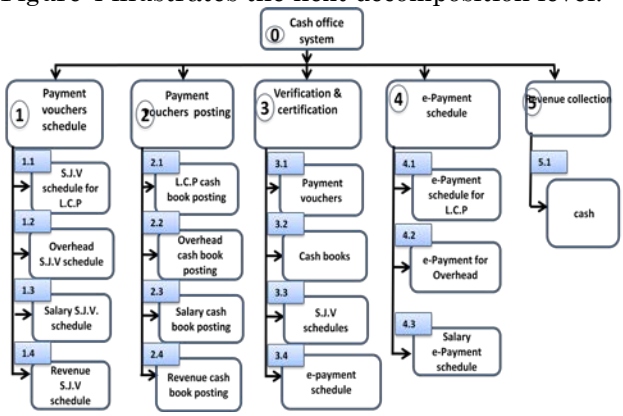

Figure 4: Second level decomposition of Cash Office Context diagram

B. *Cash Office Process: 3rd and 4th Level Decomposition*

In modeling, logical process refers to work or action that must be performed no matter how a system is implemented [1]. There are three types of logical processes: functions, events, and elementary processes. The type of process depends on where the process is in the decomposition diagram/data flow

diagram. In figure 4, each process in the second level decomposition is a set of related and ongoing activities of the Cash Office System that continuously performs its work as needed. These activities are commonly known as events sometimes called transactions that its works or actions must be completed as a whole. Thus, the events process of payment vouchers schedules need to be decomposed. Figure 5 shows the third level functional decomposition chart.

An event is triggered by a discrete input and is completed when the process has responded with appropriate outputs [3]. Therefore, each event in the decomposed functions in figure 4 needs further decomposed into elementary processes. Elementary process also known as primitive process gives detail description of how the system must respond to an event. Since elementary process are discrete, detailed activities or tasks that required complete response to an event, then in other word they are the lowest level detail depicted in a process model. Figure 6 shows the fourth level functional decomposition chart for consolidation of LCP payment vouchers.

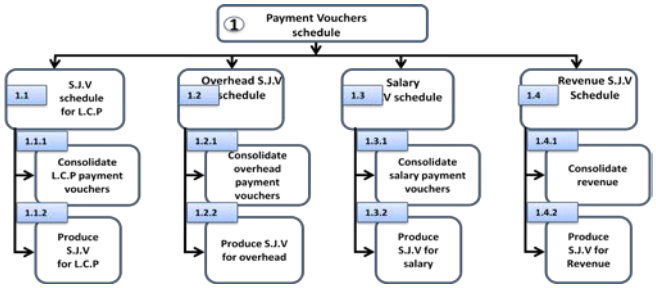

Figure 5: Third level decomposition of Cash Office Context diagram

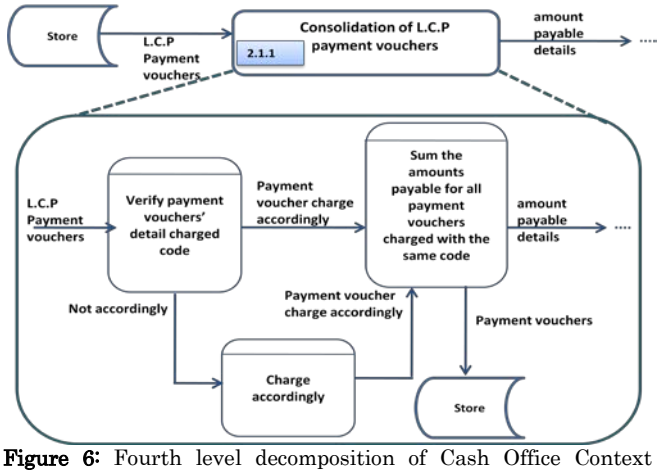

Figure 6: Fourth level decomposition of Cash Office Context diagram

6. CASH OFFICE DATA MODELING

In general, data requirements for systems development and maintenance must be modeled using entity relationship disciplines and diagrams. A single entity defined in the model may be represented graphically on several different entity-relationship diagrams.

After the requirements gathering phase, the Cash office entities were identified. In addition, existing files, reports, payment vouchers, schedules, cash books among others were studied, entities identified. The cash office system's entities can be classified as follows:

- ✓ Persons: Accountant, Cashier, Store officer, Mail officer, Revenue payee, Beneficiary
  Note: a person entity class can represent individuals.
- ✓ Objects: e-payment mandate schedule, standard journal voucher, revenue detail summary sheet, cash book, payment voucher, receipt, cash money, cheque, and third party account.
- ✓ Events: scheduling, posting, certifying, verifying, and approving.
- ✓ Concepts: Account.

Context data model is a data analysis model that includes all entities discovered as well as their relationships, no attributes. Context data model reflect new system requirements and scope. Data analysis is a technique used to improve a data model in preparation for implementation as database.

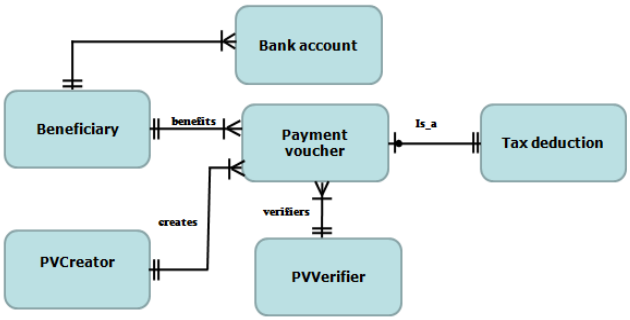

Figure 7: Context Data model for Payment Voucher Process

The process of preparing a data model for implementation is called Normalization. Requirements models should be in 3rd Normal Form as a minimum. This level of normalization is commonly described as being one where each attribute of each entity is dependent on "the key, the whole key, and nothing but the key". Where there are entities with large numbers of attributes, particularly where groups of these attributes are interdependent in their optionality or edit rules (e.g.

they must either all be null or all have assigned values), these entities should be further broken down along the patterns of these groups of attributes.

Each entity in figure 7 was normalized before representing it in entity relationship diagram. Figure 8 depicts database entity relationship diagram of the Payment Voucher process model in figure 7.

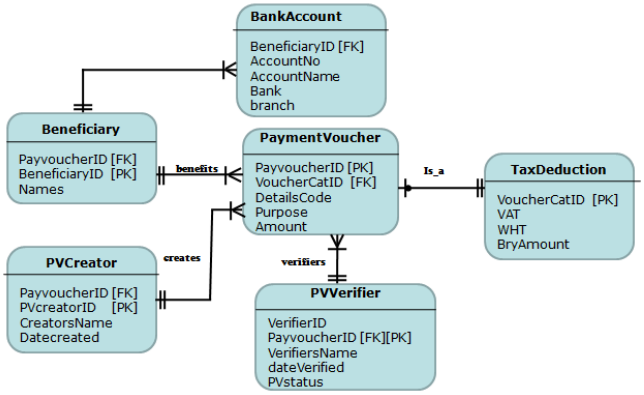

Figure 8: Database Model for Payment Voucher Process

CONCLUSION

This paper discusses the concepts of system process modeling, and using data flow diagram set of rules highlighted how to develop Process model for Bursary Cash office using data flow diagrams. Application of the set of rules reduces stress of understanding the system and eliminates errors during requirements gathering. The logical process of decomposing the Cash Office system allows proper documentation and allows developers to overcome the trouble of redesigning the system they want to develop.

REFERENCES

[1] Professor Whitten, Jeffery L. et al (2004). *Systems Analysis and Design Methodology*, 6th ed. Published by McGraw Hill, Irwin.

[2] Gary B. Shelly and Harry J. Rosenblatt (2010). *Systems Analysis and Design*, 8th ed. Published by Course Technology, Cengage learning.

[3] Alan Dennis and Barbara Haley Wixom (2003). *Systems Analysis Design*, 2nd ed. Published by John Wiley & Sons, Inc.

[4] Donald, S. and Le Vie, Jr. (2000). Understanding Data Flow Diagram. *Proceedings of the 47th annual conference on Society for Technical Communication*. Texas: Integrated Concepts, Inc.

[5] Dennis, A., Wixom, B.H. and Roth, R.M. (2006). *Systems Analysis and Design*. 3rd ed. Hoboken: John Wiley & Sons, Inc.

[6] ITMB (2003). Requirements Modeling and Specification Guidelines and Standards Version 3. 2 March 31.

[7] Jan Speelpenning et al (2001). Oracle Student Guide: Data modelling and Relational Database Design. Vol. 1, Edition 1.2, July 2001.

[8] Rosziati Ibrahim and Siow Yen Yen (2010). Formalization of the Data Flow diagram Rules for consistency Check. International Journal of Software Engineering & Applications (IJSEA), Vol.1, No.4, October 2010

[9] Rosziati Ibrahim and Siow Yen Yen (2010). An Automatic Tool for Checking Consistency between Data Flow Diagrams (DFDs). World Academy of Science, Engineering and Technology 69 2010.

[10] The Bursary Department, University of Ilorin-Nigeria. http://www.unilorin.edu.ng/index.php/support-units/bursary

[11] The Bursary Department, University of Agriculture, Abeokuta-
Nigeria. http://www.unaab.edu.ng/units/bursary.html

[12] The Bursary Department, Nasarawa State University,
Nigeria. http://www.nsukonline.net/Bursary.aspx

AUTHORS PROFILE

Friday Yakubu received the B.Tech. degree in Computer Science from Abubakar Tafawa Balewa, University, Bauchi, Nigeria, in 2003, he obtained Masters of Information Management in 2008 and M.Sc. degree in Computer Science in 2011, from A.B.U Zaria. He currently works with Ahmadu Bello University, Zaria as Software Programmer. His interests are Networking and Web Based Programming.

Barroon Isma'eel Ahmad received the B.Sc. Degree from Usmanu Danfodiyo University Sokoto Nigeria, in 2006. He joined Department of Mathematics, Ahmadu Bello University, Zaria as Graduate Assistant in 2008, and currently he is a Postgraduate Student in the same Department. His interests are Networking and Mobile Application Development.

Olugbemi, Morenike Omowumi received the B.Tech. Degree from Abubakar Tafawa Balewa, University, Bauchi, Nigeria, in 1993, she obtained M.Sc. from A.B.U Zaria in 2008. She is a Chief System Analyst with Ahmadu Bello University, Zaria as in 2006. Her interests are E-learning and Web Based Programming.

Abur, Mngohol Maria received the B.Sc. Computer Science from University of Abuja, Gwagulada in 2005, she is currently a M.Sc. student with Ahmadu Bello University, Zaria. She is Software Programmer with Ahmadu Bello University, Zaria as Software Programmer. Her interest is Web Based Programming.