FORMALIZATION OF A HOSPITAL MANAGEMENT SYSTEM

Okpor Margaret Dumebi

Department of Computer Science, Delta State, Polytechnic, Ozoro

ABSTRACT

The automation of the processing and activities of Hospital Management System (HMS) can invariably contribute greatly to the success, profitability and customer-based approach of such an organization. The use of formal specification creates a formal approach for specifying the underlying functions and properties of the system. This paper has attempted to give a formal description of the activities of a HMS system Using Zed notation. The interaction within the system is visualized using Unified Modeling Language (UML) sequence diagrams.

KEYWORDS

HMS, Z-Notation, UML.

1. INTRODUCTION

Hospital Management Software (HMS) or Hospital Information System (HIS) or Hospital Management Information System (HMIS) is software developed to facilitate easy, prudent and customer-eccentric hospital services. This includes the clinical back office and generic management processes. It integrates the entire resources of the hospital for easy organizational process [1], [7].

This software usually offers; Registration and enquiry, physician order entry, casualty and emergency management, pharmacy management system, laboratory information system, nursing and ward management, store and inventory management, duty roster management, blood bank management, ambulance services, patient referral, biomedical waste management [1]; [7]

The HMS is an all encompassing system. These systems includes: SoftClinc, Insta HMS, eHopsital Systems, Health Data Archiver, Hospital Management Software,uniwide HIMS, Medstar HIS, ProMED, Availity, evbisit, HIMS, Health.Net, SisoHIS, eHat, GeroPro, Suvarna-HIS, MD-staff, Axpert HMS [4]; [7]

Generally speaking, a specification is the expression in some formal terms and at some level of abstraction, of a collection of properties, some system should satisfy. The historical perspectives and the qualities of good specifications have been discussed in [2]; [3].

Z–Notation is a formal specification construct based on set algebra and predicate calculus for specification of computing systems [2], [3]. It has the ability of decomposing specification into small piece called schemas. By splitting the specification into schemas, we can present it piece by piece.

DOI : 10.5121/ijcseit.2017.7302

Over the years the ambiguities in software system have been addressed with appropriate formalization [2], [3]. These formalizations possibly have been devoid from most medical systems, initiating researches toward this area.

Therefore it is the intent of this research to address the aforementioned issue utilizing a formal approach.

2. MATERIAL AND METHODOLOGY

In achieving the objective of determining formal methods in ascertaining the functionalities of Hospital Management System (HMS), Z-notation and Unified Modelling Language (UML) serves as the main tools for the methodology of this work.

Z-notation uses mathematical notation to describe in a precise way the properties a software system must possess, without unduly constraining the way in which these properties are achieved [9], [6], [8]. Formal specification (Mathematical notation or Z) uses mathematical data types to model data in a system and achieve it underlining objectives. These data types are not oriented towards computer representation, but they obey a rich collection of mathematical laws which make it possible to reason effectively about the way a specified system will behave. The research uses the notation of *predicate logic* to describe abstractly the effect of each operation of the system, again in a way that enables one to reason about their behaviour.

The other main ingredient in Z is a way of decomposing a specification into small pieces called *Schemas*. By splitting the specification into schemas, we can present it piece by piece. Each piece can be linked with a commentary which explains informally the significance of the formal mathematics. In Z, schemas are used to describe both static and dynamic aspects of a system [9]

The static aspects include:

a. the state it can occupy;

b. the invariant (quantity that is unchanged by a set of mathematical operation) relationships that are maintained as the system moves from states to state.

The dynamic aspects include:

a. the operation aspect that are possible;

b. the relationship between their input and outputs;

c. the changes of state that happen.

The schema presented in this presented paper provided an avenue wherein our formal specification could be presented in fragment enabling us to associate commentaries; explaining informally the significance of the formal mathematical notation representation.

2.1 METHODOLOGY

The following are some of the basic types in Z:

{CHAR, STRING, CURRENCY, QUERY, OBJECT, COMPONENTS, BOOLEAN:: = TRUE/FALSE, DATA and OBJECT}.

The Hospital Management System (HMS) authenticates each user using his username/ID and password on the system:

System User

System User_name/ID: seq CHAR

System User_password: seq CHAR

System User: ℙ System User

Access! : Boolean User

System ∈system.access! = accepted) ᵛ (system ∉system.access ≠ accepted)

Schema 1:*SystemUser Schema*

There is no frontier to the number of registered system user the Hospital Management System can ascertain and each system user can have only one authentication and authorization privilege. Logging on, each system user must register his user ID.

System List

System: ℙSYSTEM: ℙHOSPITAL FUNCTIONALITIES

System List: SYSTEM →HOSPITAL FUNCTIONALITIES

Systems = dom list

Schema 2: *System List*

The *Systemlist* provides the hospital functionalities provided by the system.

Register Hospital functionalities

Δ System List

System? : SYSTEM

Hospital Functionalities: Hospital Functionalities

report! : REPORT

(system? ∉system˄System List′ = SystemList U {system? →Functionalities?}˄ report! = ok) ˅(system? ∉system˄systemlist′ = systemlist˄ Report! = already_known)

Schema 3: *Register Hospital Functionalities Schema*

The list braces the facility in registering hospital functionalities given that the system process does not exist previously. If the process exists previously, a report of 'already known' is returned or vice versa as the case may be.

Locate HospitalProcess

Ξ SystemList

Processes? : HOSPITAL PROCESSESS

Systemlist?ℙ HOSPITAL PROCESSES

(systemlist! = {system: systems / list (system) = process?}˅ (system ∉ systems ˄ report! = not_known)

Schema 4: *Locate Hospital process Schema*

The *Locate Hospital Process* function obtains a process type as an argument and returns a result responds.

```
AlreadyExisting Hotel Process
 Ξ SystemList
system? : SYSTEM
result! : REPORT
---
process? ∈ Processes
result! = already_known
```

***Schema 5:** AlreadyExisting Hospital ProcessSchema*

The *AlreadyExisting Hospital ProcessSchema* determines if a change to the systemlist in terms of new input process already. If it the result, will reply already_knowm otherwise not known.

```
Hospital ProcessNotAvailable
 Ξ Processlist
process: ℙHOSPITAL PROCESS
report! : REPORT
request? : HOSPITAL PROCESS
---
∀process ∈system.list (process) ≠ Reliability
report! = not_known
```

***Shema6:** Hospital ProcessNotAvailable Schema*

The schema 6 shows the error which occurs when a system user requests for a process which has not been registered in the systemlist. An error report of ‘not known’ is returned. The list is initialized at the beginning with no process or hospital process in Schema 7.

```
Hospital Initialize_List
SystemList
PROCESS: ℙHOTEL PROCESSES
---
systemlist = ∅
hotelprocess = ∅
```

***Schema 7:** Initialize_List Schema*

2.2 System Design And Unified Modeling Language (UML)

Software design immediately follows the requirements engineering phase in a software process. Software design is the translation of the requirement specification into useful patterns for implementation. Unified Modelling Language (UML) is a standard modelling language used for modelling software systems. UML was used for design of the hospital system process because UML focuses on creating simple, well documented and easy to understand software models. UML sequence diagram shows the interaction between classes (or object) in the system for each use case. The interaction represents the order of messages that are exchanged between classes to accomplish a purpose. For the system we specify the utilizing several sequence diagram specified on Figure 1.

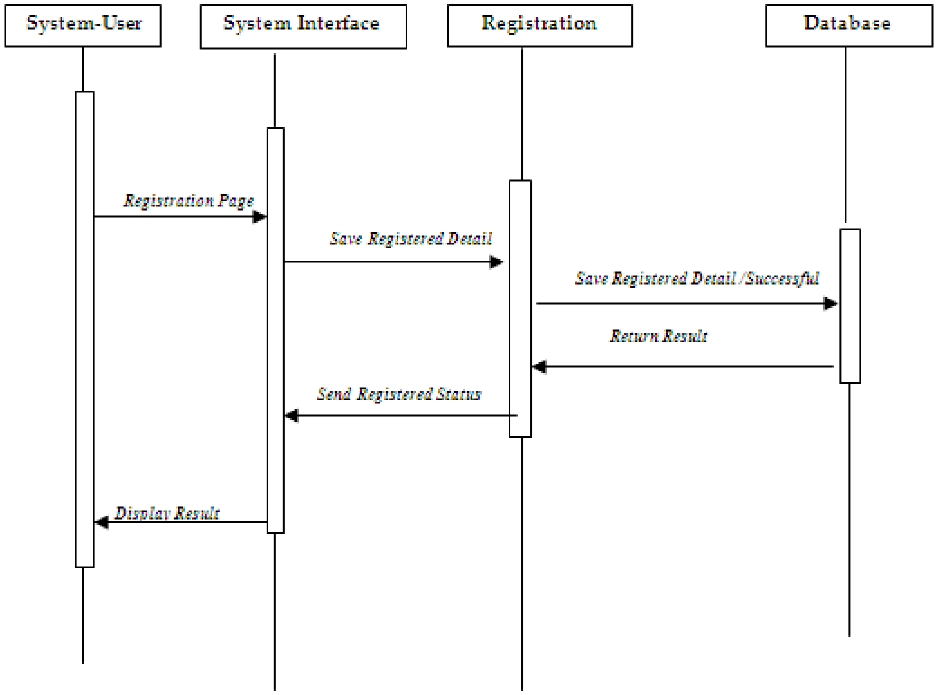


Figure 1: Registration User Sequence Diagram for HMS

Figure 1 models the sequence of steps involved in the registration of a system user. The order of appearance of the arrows indicates the order of the actions while the arrow direction indicates the direction of flow of events / results.

3. IMPLEMENTATION

The implementation of the system was handled utilizing MATLAB, for several reasons:

a. It also manipulates across varied numerical data and It integrates with numerous user interfaces.

b. Since formal specification are specified utilizing mathematical notation, MATLAB was at the frontier of all available implementation tools available due to its available mathematical ingredients.

c. It is open source

d. Most importantly, it has a large active community base

Utilizing several MATLAB tools such:

a. ***Assignment Statements*:** Assignment statement as a MATLAB tool was used in overriding Predecessor variable while the successor variable took over. The assignment statement were used in the system to override previous entries which have be saved to the system database

b. ***Case Sensitivity:*** The variable name within are system were case sensitive acceding to Matric laboratory rules. The case sensitive rules in MatLab helped us in distinctively separating our variables name

c. ***Immediate and Deferred Execution****:* When MATLAB is invoked; the user is presented with an *interactive environment*. Enter a statement, press the carriage return ("ENTER") and the statement is immediately executed. Given the power that can be packed into one MATLAB statement, this is no small accomplishment.

4. FINDINGS

Based on the ease at which the users get information through this new system, a lot are sure to be benefitted from this system. The system will:

a. Provide hospital process thereby eliminating wastage of unusual time

b. Eliminate and ratify unknown errors

c. Support multiple system processes

d. Provide user-friendliness interface

5. CONCLUSION

Formal specification is the bedrock of safety critical system which uncovers ambiguities and unwanted error from system requirement. In Nigeria and African as a whole this approach has not be implemented for most safety critical system opening the avenue for system failure with huge implication such as financial loss and untimely death. This research paper focuses on providing a sample representation of formal specification utilizing hospital proc esses as a case base. The system design was specified utilizing UML while implementation was handled exploring MATLAB. The results of the finding were listed assiduously. Formal specification is an avenue for safety critical system which must be explored in as much safety is involved.

REFERENCES

[1] Acgil (1993), Hospital Management Information System, retrieved online from http://www.acgil.com/products/hospital_management_system.htm

[2] Bowen J. P., (1988), Formal Specification in Z as a design and documentation tool. In proc. Second IEE/BCS Conference on Software Engineering, number 290 in conference publication, pages 164-168.

[3] Bowen P., (1990), Z bibliography, Oxford University Computing Laboratory.

[4] Capterra (2017), Top Hospital Management Solutions, retrieved online from http://www.capterra.com/hospital-management-software

[4] CISA: Certified Information System Auditor (2011), “Cisa review manual 2010”, Chapter 5, Pp. 305.

[5] Jiantao P. (1999), “Software Reliability”, retrieved online from http://ece.cmu.edu/koopman/Des_a 99/sw-reliability

[6] Sannella D., (1988), “A Survey of formal software development methods”, appeared in Software Engineering: A European Perspective, A. McGettrick and R. Thayer (eds.), IEEE Computer Society Press, pp 281-297, 1993.

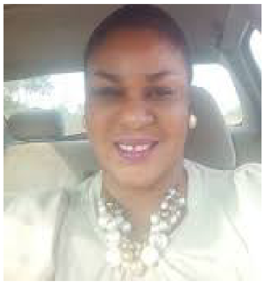
[7] SoftSuggest (2017) Best HopsitalManagenetSofwtare Solution-2017, retrived online from https://www.softwaresuggest.com/hospital-management-software

[8] Spivey J. M. (1992), “The Z Notation: A Reference Manual, 2nd Edition”, Prentice Hall International (UK) limited, United Kingdom.

[9] Spivey J. M. (1998), “The Z Notation: A Reference Manual”, Oxford, United Kingdom.

[10] Stuart R., and Norvig P. (1995), “Artificial Intelligence: A Modern Approach”, Prentice Hall (UK) International.

AUTHOR



Mrs. Margaret Dumebi Okpor has B.Sc and M.Sc degrees in Computer Science from University of Benin, Edo State, Nigeria in the years 1996 and 2012, respectively. Currently she is running a Ph.D programme in Computer Science. She has been involved in teaching Computer Science courses for over 10 years in the Delta State Polytechnic, Ozoro, Delta State, Nigeria. She was Head of Department in the Department of Computer Science in the same institution. She is a member of Computer Professionals of Nigeria (CPN). She has published several papers in local and international journals. Her research interests include machine learning, soft computing, computational intelligence and evolutional computing.