

Hospital Management System (HMS): A Formal Approach using Z-notation

Ejiofor C. I¹, Mgbeafuluike .I .J²
Chukwuemeka Odumegwu Ojukwu University ^{1,2}

Abstract

The use of formal specification creates a formal approach for specifying the underlying functions and properties of the software system. This paper has attempted to give a formal description of the activities Hospital management System(HMS) using Zed notations.

Keywords: HMS, Z-Notation

I. INTRODUCTION

Systems are a comprehensive, integrated collection of unified components fused with the sole aim of achieving a fundamental objective or say target. In generic sense system has been developed in varied fields such as: agricultural educational, mining and even the financial sector. Most of these system has been designed with the fundamental component of information technology (IT) based on its varsity in achieving underlining goals promptly and efficiently (Practo, 2015).

Overtime, IT systems has revolutionized the field of medicine, through the complementing role of IT in facilitating prompt retrieval of patient hospital records, eliminating bulky storage of files in file cabinets, thereby enhancing prompt and fast recovery. IT systems has also aided medical professionals through tele-medical processes involving querying patient files in retrieving relevant information needed for proper and efficient diagnosis. These medical based IT systems are developed interactively with the sole aim of requesting and responding to patient complaints (Adroit, 2017; Prcato, 2015).

In this fast-paced environment of medical processes, managing hospital processes can indeed become challenging. This challenge has been completed through the provision of IT based Hospital Management System (HMS). HMS is a computer or web based system that facilitates the efficient management of hospital functionalities, which if applied assiduously and efficiently will tremendous cut-down the usage of paper and other stationery with the hospitals. These systems integrate all the information regarding patients, doctors, staff and hospital administrative details (Practo, 2015; Rachita, 2017).

HMS has encouraged the prompt recording of new patient within the system. HMS stores medical details of patients including graphical images which have help in medico-legal cases. HMS has also facilitated quickly checks for available rooms and bed space, so that the transfer of patient from one ward to another is easy, prompt and efficient. HMS are built to facilitate proper record keeping pertaining to staff location, names and timing of nurses and even physician on duty (Rachita, 2017). These may also include instruction handed down for nurses pertaining to each patient. HMSs can keep track of thorough inventory including stock of supplies in the hospital like surgical instruments, medications, laboratory supplies, stationary and staff supplies. HMSs have also provided integrated data for statistical analysis purpose. This has helped in providing a statistical database for prompt access within the hospitals (Practo, 2015; Rachita, 2017). HMS is a software system and like most software system, formalization has been devoid from these systems (Michael, 2017). Formalization is the process of representing software system using mathematical notations. This has help to achieve accuracy, preciseness while eliminating ambiguities.

Therefore it is the intent of this research paper to explore Z-notation as a formal method for precisely defining the properties of Hospital Management System (HMS).

II. EXPLORED METHODOLOGY

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 (Spivey 1998, Sannella, 1998 and Spivey, 1992). 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. We use the notation of predicate logic to describe abstractly the effect of each operation of our system, again in a way that enables us 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 (Spivey 1998). The static aspects includes: the state it can occupy, the invariant (quantity that is unchanged by a set of mathematical operation) relationship that are maintained as the system moves from states to state. The dynamic aspect includes: the operation aspect that are possible, the relationship between their input and outputs and the changes of state that happen.

The schema presented in this presented paper provided an avenue wherein our formal specification

System_User
System User_name/ID: seq CHAR System User_password: seq CHAR System User: \mathbb{P} System User Access! : Boolean User System \in system.access! = accepted) \vee (system \notin system.access \neq accepted)
Schema 1: System_User Schema

could be presented in fragment enabling us to associate commentaries; explaining informal the significance of the formal mathematical notation representation.

III. APPLIED METHODOLOGY

Hospital Management System (HMS) was formalized adopting and applying the notation of Z with its basic types. 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) authentic each user using individual physician username/ID and password on the system.

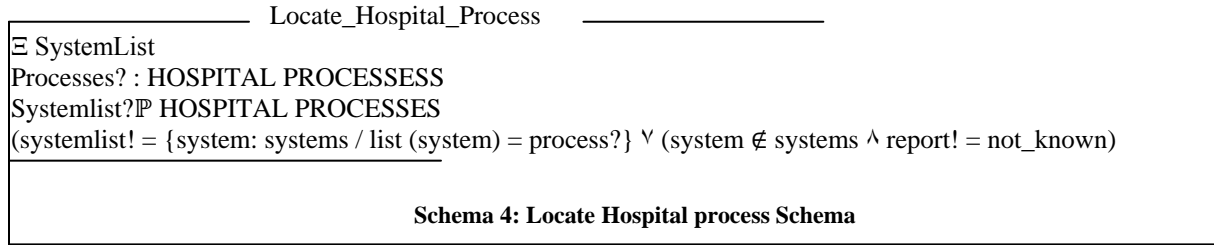
The sole system user is assigned to register physician with no frontier to the number of registered physician utilizing the Hospital Management System (HMS). HMS ascertains and each system user can have only one authentication and authorization privilege. Logging on, each physician must register it's his user ID.

System_List
System: \mathbb{P} SYSTEM: \mathbb{P} HOSPITAL FUNCTIONALITIES System List: SYSTEM \rightarrow HOSPITAL FUNCTIONALITIES Systems = dom list
Schema 2: System List

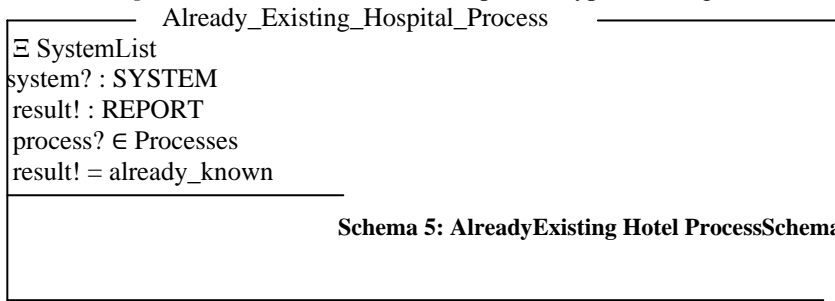
The *Systemlist* provide the hospital holistic functionalities provide by the system. These functionalities determine processes available across the system.

Register_Hospital_Functionalities
Δ System List System? : SYSTEM Hospital Functionalities: Hospital Functionalities report! : REPORT (system? \notin system \wedge System List' = SystemList U {system? \rightarrow Functionalities?} \wedge report! = ok) \vee (system? \notin system \wedge systemlist' = systemlist \wedge Report! = already_known)
Schema 3: Register Hospital Functionalities Schema

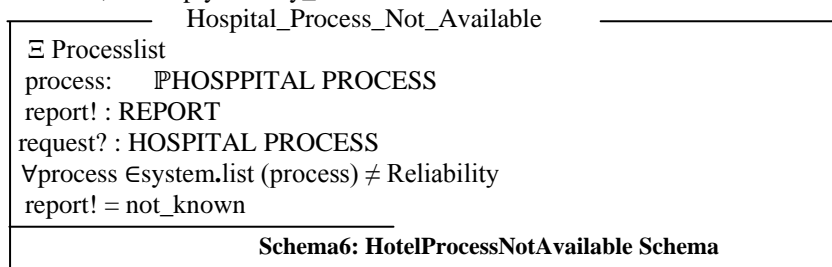
The list braces the facility in registering hospital functionalities given that the system process does not exist previously. If the process exists previously, a report of 'already known' is returned or vice versa as the case may be.



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



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



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.

IV. IMPLEMENTATION TOOL

The implementation of the Hospital Management System (HMS) will be better achieved using object oriented programming tool due to schema structure mimicking object orientation. It is hoped on fully implementation the following objectives will be achieved

- 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

V. CONCLUSION

Formal specification is the bedrock of most software system which uncovers ambiguities and unwanted error from system requirement.. This research paper focuses on providing a sample representation of formal specification utilizing Hospital processes as a

case base. The forecasted benefit should prompt fast and assiduously implementation

REFERENCES

- [1] Adroit (2017), eHospital System, retrieved online from <https://www.adroitinfosystems.com/Products/ehospital-systems>.
- [2] Practo, (2015), Hospital Management System, retrieved online from <https://doctors.practo.com/hospital-management-system/> 10 October, 2017
- [3] Rachita N. (2017), Hospital System, <http://itprojectshub.com/documentation/hospital-management-system.docx>
- [4] Jiantao P. (1999), “Software Reliability”, retrieved online from http://ece.cmu.edu/koopman/Des_a_99/sw-reliability/
- [5] Michael J. (2017), Formalism and Informality in Software System, retrieved online from https://link.springer.com/chapter/10.1007/3-540-46019-5_31
- [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] Spivey J. M. (1992), “The Z Notation: A Reference Manual, 2nd Edition”, Prentice Hall International (UK) limited, United Kingdom.
- [8] Spivey J. M. (1998), “The Z Notation: A Reference Manual”, Oxford, United Kingdom.
- [9] Stuart R., and Norvig P. (1995), “Artificial Intelligence: A Modern Approach”, Prentice Hall (UK) International.