Journal of Physical Science and Innovation **Volume 7, No. 2, 2015**
ISSN: 2277-0119

SCENARIO-BASED DYNAMIC AND STATIC SEPARATION OF DUTY

Nura M. Shagari[1], Buhari Wadata[2], Abubakar Ibrahim[3], and Salisu Modi[4]
[1,2,4,5] Department of Computer Science, Sokoto State University
[3]Department of Computer Science, Sokoto State Polytechnic
Email:abunbba@yahoo.com

ABSTRACT

Role-based access control policies allow access to the resources based on the role the user has within the system and the roles specifies what accesses are allowed to users in a given roles. This paper critically analysed role-based access control for a scenario (Medical Centre). The goal was to access how dynamic and static separations of duty are extracted in real life scenario. RBAC model standard 2004 was adopted for the definition of basic RBAC system elements from the scenario using role engineering technique. The RBAC system was found to be a promising access control model that ensures data integrity, confidentiality, and availability and lower the costs of security administration.

Keywords: *Role, Access Control, Role Engineering, Data Integrity and Confidentiality.*

INTRODUCTION

As organizations increase reliance on database systems for day-by-day operation and decision making, the security of data managed by the organization becomes imperative and crucial. Unauthorized access, incorrect modifications of data as well as unavailability of data affect not only a single user or application, they also have disastrous consequences on the entire organization. An important requirement of any information management system is to protect data and resources against unauthorized users as well as unauthorized or improper modifications, while at the same time ensuring its availability to legitimate users (di Vimercati, et al. 2005). Therefore, the need to enforce protections to ensure that every access to the system and its resources is controlled and only authorized access take place is of paramount importance. This activity is known as *access control*. Access control regulates all access to the system by the users to ensure all access is authorized according to some specified policy. Role-based access control (RBAC) is a security policy that is widely accepted in the field because it greatly lowers the cost and complexity of securing large network and web-based systems (Ferraiolo, et al. 2007). RBAC is an alternative to

traditional discretional (DAC) and mandatory access control (MAC) policies that is receiving increase acceptance for commercial application (Gligor, 1996). The motivation behind role-based access control development is its flexibility for allowing specification and enforcement of policies that maps naturally to an organization's policy and structure (Samarati and de Vimercati, 2001). While mandatory access control (MAC) are appropriate for multilevel secure military applications, Discretionary access control (DAC) are often considered as meeting the need of industries and civilian government, however the security policies in DAC are not appropriate for most organizations as each organization has unique security requirement. In DAC, subject with certain access permission is capable of passing that privilege to another subject i.e. granting and revocation of access privilege is left at the discretion of individual users. Subjects can grant and revoke privilege to other users under their control without the consent of the system administrator. However, many organizations are the owner of the system objects as well as the programs that access it (Ferraiolo and Kuhn, 2007). The control is often based on employee's function and not on data ownership. Hence it is inappropriate to allow users to pass their privilege to other users in these organizations. In addition, mandatory access is not appropriate in these organizations because its application is unbending and this makes it unsuitable for organization with dynamic roles. It is easier to achieve access control management, authorization management in information systems by using RBAC model than by DAC and MAC (Zhu, et al. 2012).

1. RBAC Model

Role-based access control is a non-discretionary access control in which the system administrator allows role's permissions to user, by defining user, role and permission. User access resource based on permissions attached to the role which the user belongs. Users who are granted role in the system manage their works with their role permission. The basic elements of RBAC model are User, Role and Permission.

- User is a person who uses the system or application program within the system.
- Role represents functional responsibilities within the organization. The system administrator defines the roles, a combination of obligations and authority in organizations and assigns them to users.
- Permission determines the access right of the role.

RBAC model consists of Role-Permission relationship, User-Role relationship and Role-Role relationship. User-Role relationship represents which user is assigned to perform what kind of role in the organization.

Role-based access control standard (Standard, 2004) consists of two parts:

- A reference model, which specifies sets of basic RBAC elements and functions
- A system and administration functions specification, which specifies the RBAC system's operations and functions.

These models are each divided into four component parts that correspond to the four RBAC components:

- Core RBAC
- Hierarchical RBAC
- Static separation of duty (SSD) relations, and
- Dynamic separation of duty (DSD) relations.

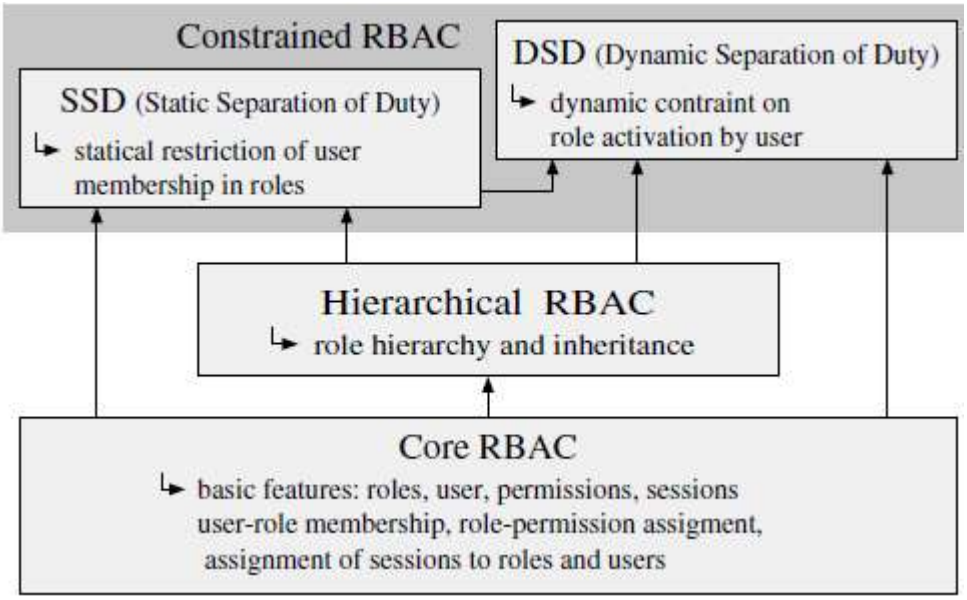

Figure 1: Overview of RBAC model (Dridi, et al. 2004)

Core RBAC

Core RBAC allows permission to be assigned to the roles and assigned roles to individual users so they obtain permission. It consist the following functions and relations

- USERS, ROLES, OPS and OBS (users, roles, operations and objects, respectively).
- UA ⊆ USERS X ROLES, a many-to-many mapping user-to-role assignment relation
- PA ⊆ PRMS X ROLES, a many-to-many mapping permission-to-role assignment relation
- Assigned_permissions (r: ROLES)→ 2^{PRMS} , the mapping of role r onto a set of permissions.
- PRMS = $2^{(OPS\ X\ OBS)}$, the set of permissions
- Assigned user : (r : ROLES) → 2^{USERS}, mapping role r on to set of users;

- Op(p :PRMS) → $\{op \subseteq OPS\}$, the permission-to-operation mapping, that gives the set of operations associated with the permission p
- Ob (p: PRMS)→ $\{ob \subseteq OBS\}$, the permission-to-object mapping, that gives the set of objects associated with permission p.
- SESSIONS, the set of sessions

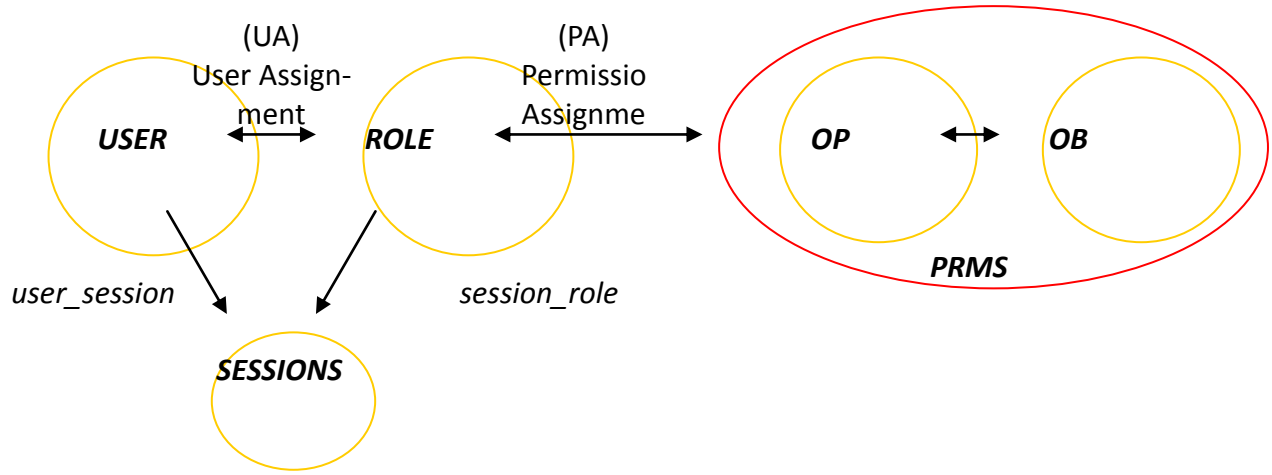

Figure 2: Core RBAC (ANSI, 2004)

Hierarchical RBAC

Hierarchical RBAC defines inheritance relationship among roles to reduce administration costs. The standard describes the general role hierarchies

- RH ⊆ ROLES X ROLES is a partial order on ROLES called inheritance hierarchy, written as ≥ where $r_1 \geq r_2$ only if all permission of r_2 are also permissions of $r_1$and all users of r_1 are also users of r_2,that is $r1 \geq r2 \Rightarrow authorized\_permissions(r2) \subseteq authorized\_permission(r1)$
- Authorized_users (r: ROLES)→ 2^{USERS}, the mapping of role r onto a set of users in the existence of a role hierarchy. Formally $authorized_{users(r)} = \{u \in USERS | r' \geq r, (u, r') \in UA$
- Authorized_permissions (r: ROLES) → 2^{PRMS}, the mapping of role r onto a set of permissions in the existence of role hierarchy.

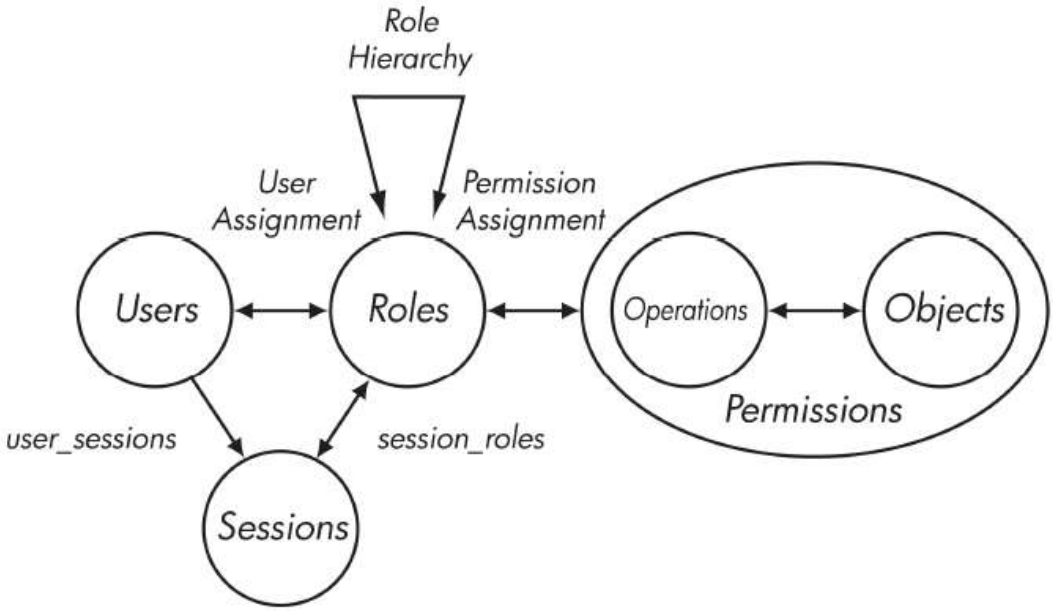

Figure 3: Hierarchical RBAC (Coyne C., 2009)

Constrained RBAC

The constrained RBAC component has two types of constraints: static separation of duty (SSD) and dynamic separation of duty (DSD). SSD constraint restricts the roles a user can be authorized for and DSD constraint bounds the roles that a user can activate in one session.

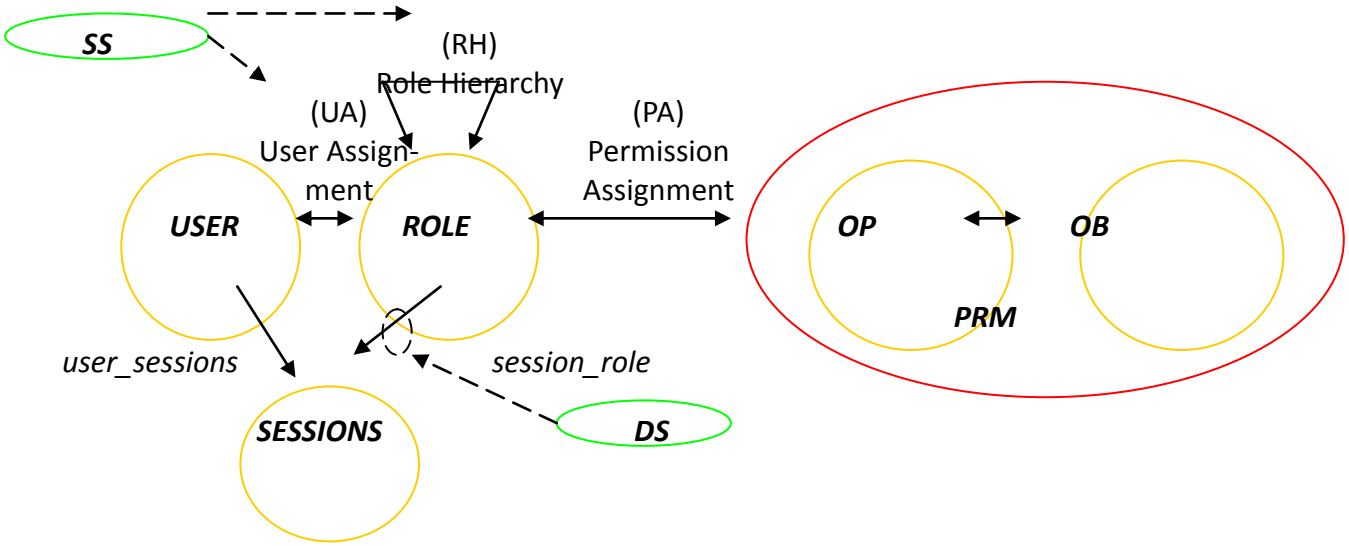

Figure 4: Constrained RBAC (Coyne C., 2009)

2. Scenario Schematic

A simple medical centre scenario was use for the definition of basic RBAC elements.. This scenario was used for definition of role-to-role relationship, separation of duty among roles and permission assignment to the role.

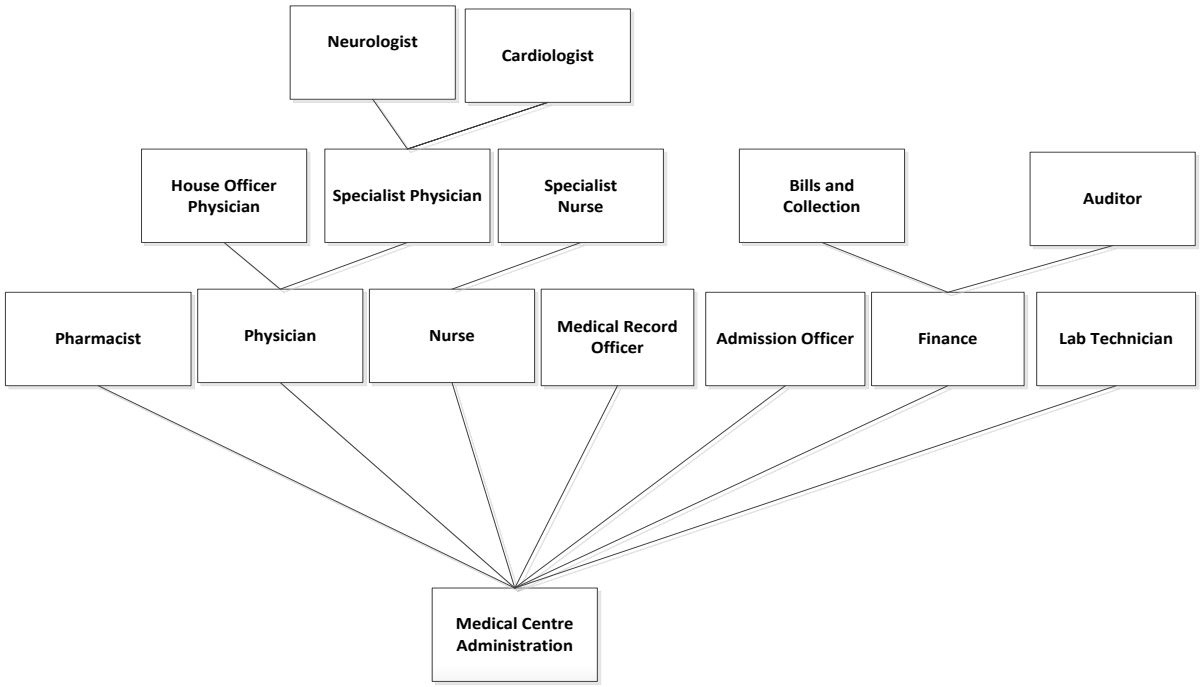

Figure 5: Scenario considered

Roles

- Physician – Nurse specialist – Auditor
- Pharmacist – Nurse – Patient
- Medical record officer – Cardiologist – Admission officer
- Billing and collection officer – Neurologist – Laboratory technician
- Hospital administrator

3. Role Engineering

To precisely define roles, tasks and operations, the study uses role engineering techniques. The fundamental issue in role-based access control is the existence of set of roles that accurately define the activities, functions and responsibilities within the organization. Definition of role, task, and operation must be accomplished before the benefit of RBAC can be fully attained. Consequently, the definition of role is an essential requirement of engineering process (Coyne, 1996). The overall essence of role engineering is to ensure accurate, efficient and complete definition of roles.

Role Identification

The concept of role engineering (RE) is an approach to defining roles and assigning permission to the roles. The role engineering tends to capture all organization's business rules in relation to access control and reflects these rules in defining, naming, structuring

and constraining valid sets of roles. Role engineering often tends to include all components of RBAC model with the exception of assignment of users to the roles. The RBAC model previously described contains the following components

- Core RBAC where permissions are assigned to roles and users are mapped to roles
- Hierarchical RBAC which defines the inheritance relationship among roles
- Constrained-RBAC where constraints among roles and assignment may be defined.

The components of RBAC model necessary to be defined as part of role engineering are (Sandhu et al., 1996):

- Roles
- Permissions
- Constraints
- Hierarchies

Identification of roles, permissions, constraints and role hierarchy can be performed following the steps described in the figure below.

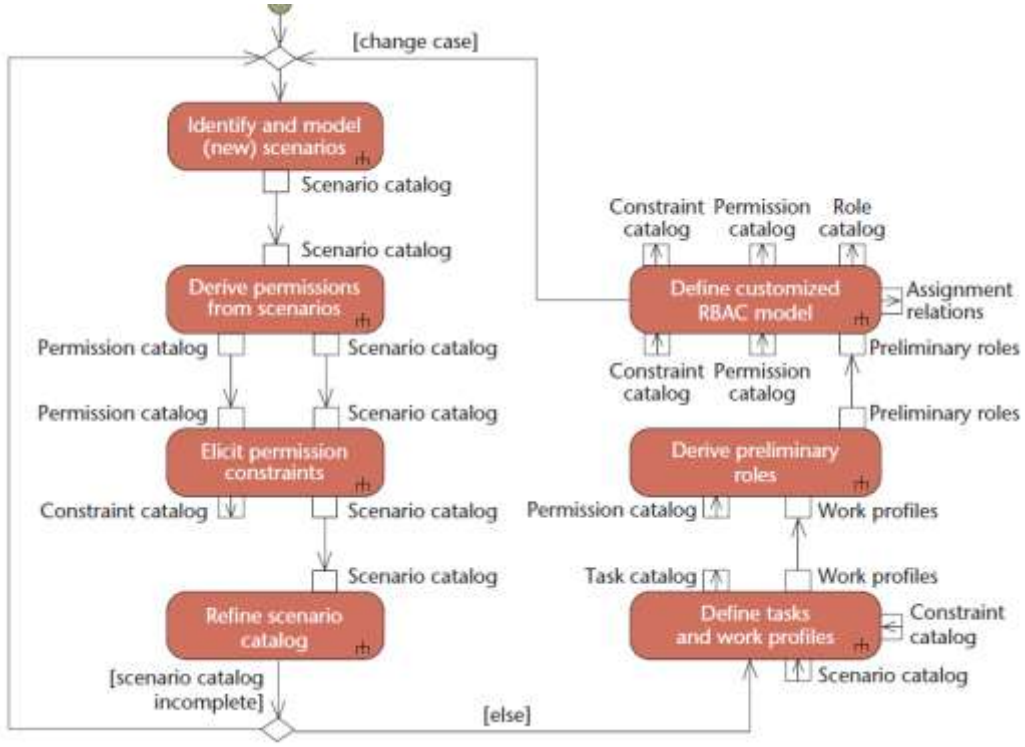

Figure 6: Scenario driven role-engineering process (Strembeck, 2010)

- Scenario catalog comprises all usage scenarios for the system under consideration such Medical Centre in our case
- The permissions catalog consists all permissions identified for a system
- The task catalog includes the tasks that human users or other subjects perform

- Work profile catalog consists of different work profiles. Each work profile is intended to be a complete description of all tasks that a specific type of subject must or can perform.
- Constraints catalog includes constraints that must be enforced on permissions, roles or assignment relations

By implementing the above strategies, the following descriptions and assignments were drive

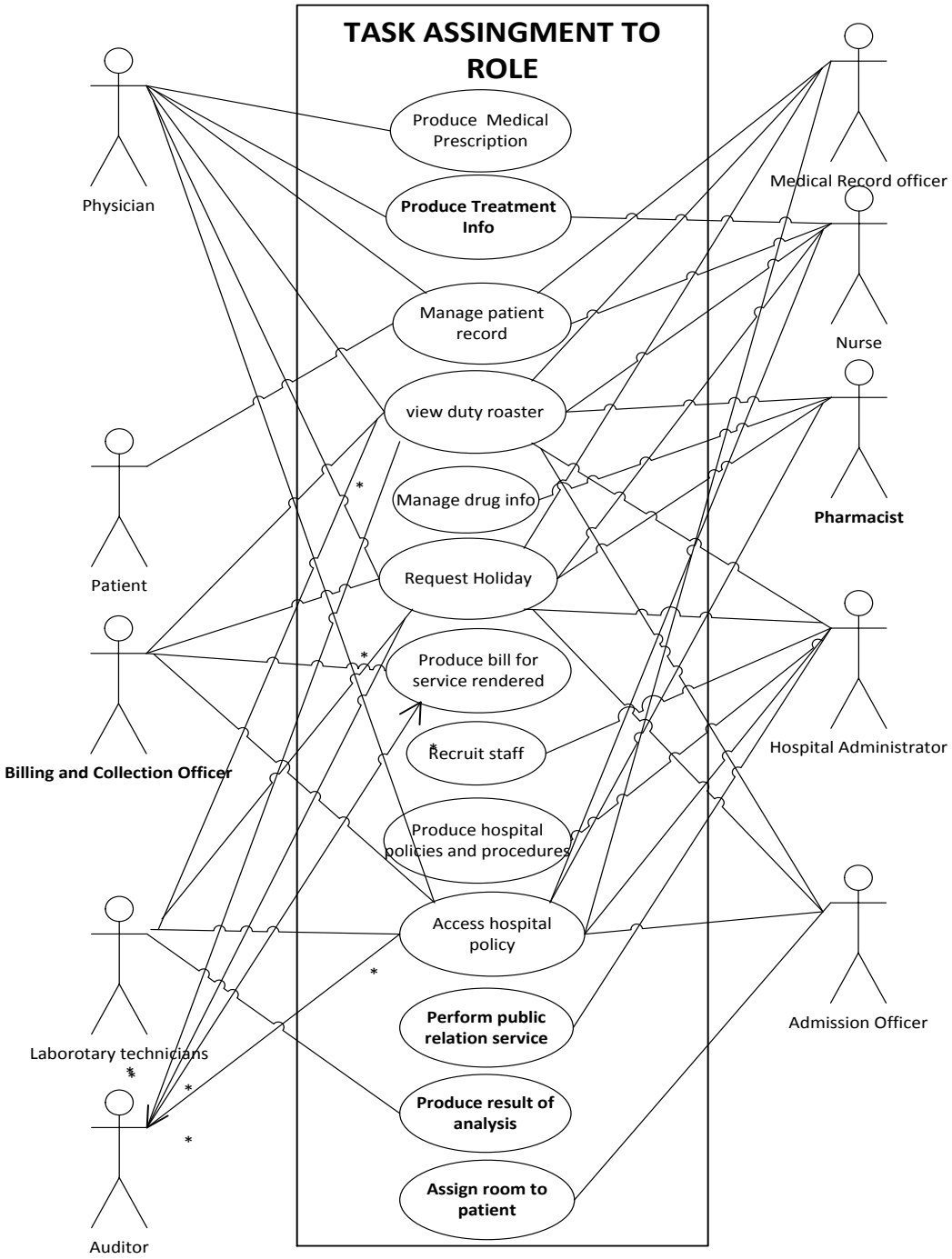

Figure 7: Task to role assignment

Table 1.Task Catalog

Use Case	Description
Produce Medical Prescription	Give access to only one role specified for this task. No other role are allowed to do this task
Produce Treatment Info	Give read, write, edit and delete access to physician to describe treatment, and read access to Nurse to read for patient care and treatment
Manage Patient Record	Give access to only four roles specified for this task. No other role is allowed to do this task. The physician should be given read, write and edit access to the record. The medical record officer should be given read, write, edit and delete access to the record. While Nurse will be given read access to the record for treatment. Give read access to patient to view his record
View Duty Roster	Give access right so that all nine roles are able to view duty roster
Manage Drug Info	Give access to only one role specified for this task. No other role are allowed to do this task
Request Holiday	Give access right so that all nine roles are able to request holiday
Produce Bill for Service Rendered	Give access to only two roles specified for this task. No other roles are allowed to do this task. Give read and write access to Bill and Collection officer. Auditor is given read, edit and delete access to overseas financial records
Produce Hospital Policies and Procedures	Give access to only one role specified for this task. No other role are allowed to do this task
Access Hospital Policy	Give access right so that all nine role are able to read hospital policy and hospital administrator in addition with write , edit and delete access
Produce Result of analysis	Give access to only one roles specified for this task. Give physician read access. No other role are allowed to do this task
Assign Room to Patient	Give access right to only one role specified for this task. No other role are allowed to this task
Perform Public Relation Service	Give access to only one role specified for this task. No other role are allowed to this task
Recruit Staff	Give access to only one role specified for this task. No other role are allowed to this task

Permission Assignments

RBAC MATRIX table maps set of permissions to the roles according to the principle of least privilege. Each role is given access to the resource according to its minimum privilege to do the job.

RBAC Matrix

RBAC matrix determines the rights of the roles. The left hand side column contains the roles of the system. The top row contains the resources while the cells contain the access rights according to the roles. The table below shows the access control matrix for the scenario considered.

Table 2. Access control matrix (permission catalogue)

Access control matrix (permission catalogue)

Read = R, Write = W, Delete = D and Edit = E

Role Hierarchy

Role hierarchy defines inheritance relation among roles in order to reduce security administration cost. The inheritance relationship among roles is determined for example as previously described in RBAC model, role r_1 inherit role r_2 if and only if all the permissions

	Medical prescription	Treatment Record	Patient Record	Time Tabling	Drug Man-agement	Holidays	Financial Record	Recruitment website	Policies	News	Test Result	Patient admission
Physician	RWED	RWED	RWE	R	–	RWED	–	–	R	R	R	–
Medical Record Officer	–	–	RWED	R	–	RWED	–	–	R	R	–	–
Nurse	R	R	R	R	–	RWED	–	–	R	R	–	–
Pharmacist	R	–	–	R	RW	RWED	–	–	R	R	–	–
Billing and Collection Officer	–	R	–	R	–	RWED	RW	–	R	R	–	–
Hospital Administrator	–	--	–	RWED	–	RWED	–	RWED	REWD	RWED	–	–
Admission Officer	–	–	–	R	–	RWED	–	–	R	R	–	RWED
Laboratory Technicians	–	–	–	R	–	RWED	–	–	R	R	RWED	–
Patient	–	–	R	–	–	–	–	–	--	–	–	–
Auditor	–	–	–	R	–	RWED	RED	–	R	R	–	–
Finance	–	–	–	R	–	RWED	R	–	R	R	–	–

of r_1 are also permission of r_2 and all authorized users of r_1 are also authorized users of r_2.
$r1 \geq r2 \Rightarrow authorized\_permissions(r2) \subseteq authorized\_permission(r1)$ and $authorized_{users(r)} = \{u \in USERS | r' \geq r, (u, r') \in UA$.
A seen from the scenario schematics above, it was observed from the hierarchical structure of the scenario that there exist inheritance relationships between the following set of roles

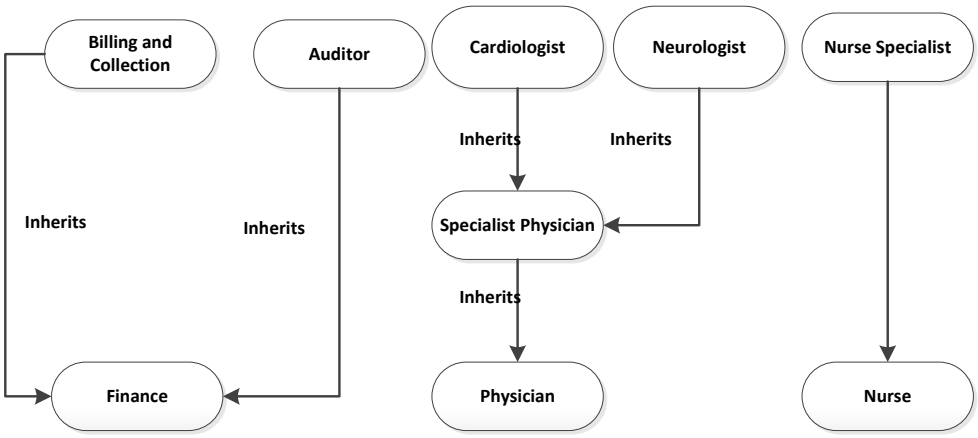

Figure 8: Task to role assignment

$R_C \geq R_S \geq R_P$, $R_N \geq R_S \geq R_P$, $RNS \geq R_U$, $R_{BC} \geq R_F$, and $R_{AD} \geq R_F$

Where Role cardiologist $= R_C$

Specialist Physician $= R_S$

Physician $= R_P$

Neurologist $= R_N$

Specialist Nurse $= R_{SN}$

Nurse $= R_U$

Billing and Collection $= R_{BC}$

Finance $= R_F$

Auditor $= R_{AD}$

Physician role: R_P

Let from the **table 2** above **RWED**-read, write, edit and delete medical prescription permission group $= P_1$

RWED-read, write, edit and delete treatment record permissions $= P_2$

RWE-read, write and edit patient record permissions $= P_3$

R-read timetable $= P_4$

RWED-read, write, edit and delete holidays $= P_5$

R-read policy $= P_6$

R-read News $= P_7$

R-read test $= P_8$

Role physician R_P permission groups $= \{P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8\}$

Since there exist inheritance relationship between role physician R_P and role physician Specialist R_S **(i.e. $R_S \geq R_P$).** Consequently, the permissions of R_S include all the permissions of R_P plus other additional permissions.

Role Physician Specialist (R_s)

Role R_S might have the following supplementary permissions

R–read special patient record = P_9

R–read special treatment =P_{10}

R–read special Prescription =P_{11}

$R_P = \{P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8\}$

$R_S = \{P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9, P_{10}\ P_{11}\}$

$-R_S = \{R_P, P_9, P_{10}\ P_{11},\}$

Role–Cardiologist R_C

Role Cardiologist inheritance relationship $R_C \geq R_S \geq R_P$ i.e. all permissions of R_P are inherited by R_S, and similarly all permissions of R_S are inherited by R_C

Role cardiologist might be authorized to observe the following additional permissions

WED–write, edit and delete cardiac Patient Record =P_{12}

WED–write, edit and delete cardiac treatment record =P_{13}

WED–write, edit and delete cardiac prescription =P_{14}

$R_P = \{P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8\}$

$R_S = \{P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9, P_{10}, P_{11}\}$

$-R_S = \{R_P, P_9, P_{10}\ P_{11},\}$

Therefore role R_C permissions = $\{R_P, R_S, P_{12}, P_{13}, P_{14}\}$

Role Neurologist (R_N)

Role Neurologist inheritance relationship, $R_N \geq R_S \geq R_P$. All permissions of R_P are inherited by R_S, and similarly all permissions of **RS** are inherited by R_N. The role Neurologist R_N is assumed to have the following additional permissions

WED– write, edit and delete neuron Patient Record =P_{15}

WED– write, edit and delete neuron treatment record =P_{16}

WED– write, edit and delete neuron prescription =P_{17}

$R_P = \{P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8\}$

$R_S = \{P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9, P_{10}\ P_{11}\}$

$-R_S = \{R_P, P_9, P_{10}\ P_{11},\}$

The role Neurologist permissions are:

$R_N = \{P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9, P_{10}\ P_{11}, P_{15}, P_{16}, P_{17}\}$

$R_N = \{R_P, R_S, P_{15}, P_{16}, P_{17}\}$

Role Nurse (R_U) permissions

Let **R**–read medical prescription permission = X_1

R–read treatment record = X_2

R–read patient record = X_3

R–read timetable = X_4

RWED– read, write, edit and delete holidays = X_5

R–read policy = X_6

R–read news = X_7

$R_U = \{X_1, X_2, X_3, X_4, X_5, X_6, X_7\}$

Role Nurse Specialist (R_{NS}) permission

There exist inheritance relationship between role Nurse R_N and role Nurse Specialist R_{NS} i.e. $R_{NS} \geq R_U$. All the permissions of role Nurse are inherited by role Nurse Specialist. Consequently, R_{NS} permissions include all the permissions of role R_U plus other additional permissions. Role R_{NS} additional permissions might include

R–read special treatment record = X_8

R–read special patient record = X_9

Therefore, $R_{NS} = \{X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_9\} \rightarrow \{R_U, X_8, X_9\}$

Role Finance (R_F) permissions

R–read timetable = Y_1

RWED– read, write, edit and delete holidays = Y_2

R–read financial record = Y_3

R–read policy = Y_4

R–read news = Y_5

$R_F = \{Y_1, Y_2, Y_3, Y_4, Y_5\}$

Role Bill and Collection (R_{BC}) permissions

The role R_{BC} inherits all the permissions of role finance R_F, formally $R_{BC} \geq R_F$. Therefore, the role R_{BC} permissions include all the permissions of R_F plus additional permission privileges. This additional permission for role R_{BC} includes

W–write to financial record = Y_6

As a result role R_{BC} permissions = $\{Y_1, Y_2, Y_3, Y_4, Y_5, Y_6\}$ and this implies $R_{BC} = \{R_F, Y_6\}$

Role Auditor (R_{AD}) permissions

Role auditor inherits role finance $R_{AD} \geq R_F$. So, R_{AD} permissions = R_F permissions + additional privileges. Role R_{AD} supplementary permission includes

ED–Edit and delete financial record permissions = Y_7. With this extra permission, the role R_{AD} permissions became $\{Y_1, Y_2, Y_3, Y_4, Y_5, Y_7\}$

$R_{AD} = \{R_F, Y_7\}$.

Constrained–Role

To prevent mutually exclusive roles assigned to a single user. It was observed from the scenario, the mutually exclusive roles are **Billing and Collection** and **Auditor**, and the roles **Cardiologists** and **Neurologists.** The role **Billing** and **Collection** is given **READ** and **WRITE** permissions to financial record while the role **Auditor** is given **READ**, **EDIT** and **DELETE** permission to the financial record. This help to prevent the possibility of fraud, since if **Billing** and **Collection** were given all the access privileges he/she might likely manipulate financial record and similarly if **Auditor** is given all access to the financial record. This is called static separation of duty. The roles cardiologist R_C and neurologist R_N are considered mutually inclusive since there exist conflict of interest between them. So, these mutual exclusive roles cannot be activated by a single user at the same time and on the same objects. This is called dynamic separation of duty as depicted in figure 9 below

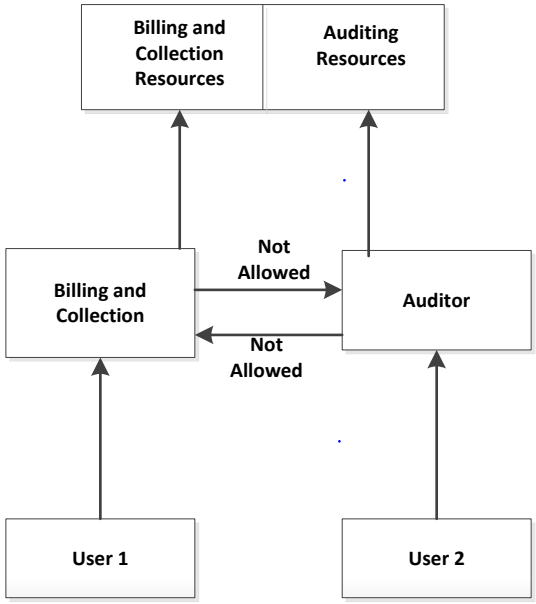

Figure 9: Mutual Exclusive roles

CONCLUSION

Role-based access control implementation in real life scenario is a challenging task. Some of the daunting challenges experience in this work include, documenting organizational policies such as Medical Centre scenario so that each role is given minimum privilege permissions to access resource. If these policies are poorly documented, the RBAC system will become more of a hindrance rather than blessing. It also posed challenge in the sense that the determination of right permission requires intimate knowledge of how permissions are being granted, why and what operations are associated with these permissions and roles. Ability to identify mutual exclusive roles and ensuring that no single user is given full control of entire business process needs careful examination of permissions that are dependent on each other and on what roles those permissions are assigned to. Thus, these roles assigned with mutual exclusive permissions are considered as mutual exclusive roles. Consequently, the security administration needs extra careful in assigning users to these roles to avoid one man control over the entire business process. The need to recognize inheritance relationship among roles and their full implementation requires skills and expertise of knowing in what sense the roles are hierarchically related. Separation of duty avoids possibilities of assigning mutually exclusive roles to a single user which prevents the chances of committing fraud in Medical Centre. This research dealt by identifying conflicting roles and ensures that no single user is allowed to assume these two roles at the same time and on the same object.

REFERENCES

Coyne, E.J. (1996) "Role Engineering", *Proceedings of the First ACM Workshop on Role-Based Access Control* ACM, pp. 4.

Vimercati, Sabrina De Capitani, Samarati, P. and Jajodia, S. (2005) "Policies, Models, and Languages for Access Control" in *Databases in Networked Information Systems* Springer, pp. 225-237.

Dridi, F., Muschall, B. and Pernul, G. (2004) "Administration of an RBAC System", *System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on* IEEE, pp. 6 pp.

Ferraiolo, D., Kuhn, D.R. and Chandramouli, R. (2007) *Role-Based Access Control,* Artech House Boston.

Gligor, V. (1996) "Characteristics of Role-Based Access Control", *Proceedings of the first ACM Workshop on Role-Based Access control* ACM, , pp. 10.

Samarati, P. and de Vimercati, S.C. (2001) "Access Control: Policies, Models, and Mechanisms" in *Foundations of Security Analysis and Design* Springer, , pp. 137-196.

Sandhu, R.S., Coyne, E.J., Feinstein, H.L. and Youman, C.E. (1996) "Role-Based Access

Control Models", *Computer,* vol. 29, no. 2, pp. 38-47.

Strembeck, M. (2010) "Scenario-Driven Role Engineering", Computer and Reliability Societies, IEEE, Vol. 8, no. 1, pp. 1540-7993.

Standard, R. (2004) "INCITS 359-2004", *ANSI INCITS,* , pp. 359-2004.

Zhu, R., Ning, J. and Yu, P. (2012) "Application of Role-Based Access Control in Information System", *Wavelet Active Media Technology and Information Processing (ICWAMTIP), 2012 International Conference on* IEEE, pp. 426.

Reference to this paper should be made as follows: Nura M. Shagari, *et. al.,* (2015), Investigations of Structural and Electronic Properties of SIII - Bi Semiconductor Binary Compounds using Density Functional Theory. *J. of Physical Science and Innovation,* Vol. 7, No. 2, Pp. 82 – 96.