Role-Based Access Control (RBAC) Role Engineering Process

Version 3.0

Developed For:
The Healthcare RBAC Task Force

Developed by:
Science Applications International Corporation (SAIC)
11 May 2004

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 Introduction

A Healthcare Role-Based Access Control (RBAC) Task Force (TF) composed of individuals knowledgeable in healthcare work profile is engaged in a collaborative effort to define common industry-wide roles capable of supporting health information systems. The ultimate goal of the RBAC Task Force is to formalize this shared effort through an ANSI-approved healthcare role standard. To meet its goals, the Healthcare RBAC Task Force will create a harmonized list of Healthcare permissions along with associated work profile representations as a work product for use in developing a new RBAC standard.

This RBAC Role Engineering Process document describes the methodology that the Task Force will follow in pursuing its goal. The RBAC Role Engineering Process describes the organization of the Task Force and the mechanisms, processes, and products that will be used to create, harmonize, and report Task Force efforts.

2 Motivation

> ***"Should this person (or a person who performs this job function) typically be allowed to access this type of data?"***

RBAC is a method to control access to resources on an information system. It was developed to overcome the complexities of managing individual user permissions and their assignments. The RBAC TF effort is motivated by concurrent efforts to:

- Simplify authorization management
- Reduce administrative costs
- Improve security
- Enhance partner interoperability
- Enable new network-level RBAC services
- Improve service to members/clients/patients

The operational benefits of RBAC have long been recognized since it simplifies the complexity of managing user permissions in large networked environments, thus providing reduced administrative cost and time. RBAC is neither a new concept, nor unique to the healthcare industry. Its implementation within healthcare systems, however, has been a challenge with a vast array of healthcare personnel roles and tasks, as well as the variety of non-homogeneous commercial and proprietary environments.

RBAC is critically important to the security aspects of healthcare organizations. Additionally, there is a growing management and security demand for RBAC to be implemented in healthcare systems.

3 Healthcare and Enterprise RBAC Task Forces

The Healthcare RBAC TF will initially be composed of representatives from each of four collaborating healthcare organizations including Kaiser Permanente (KP), Department of Veterans Affairs (VA), Department of Defense (DoD), and Indian Health Service (IHS), as shown in Figure 1. Other healthcare organizations may join the collaboration in the future.

Each participating organization will establish its own Enterprise RBAC TF. Each Enterprise RBAC TF will be composed of knowledgeable individuals (health care providers, system developers, security experts, etc) as needed within that organization. Representatives from each Enterprise RBAC TF will participate in the Healthcare RBAC TF. Standards Development Organizations (SDO) will participate as advisory members. These include American Society for Testing and Materials (ASTM), Health Level Seven (HL7), and National Institute of Standards and Technology (NIST).

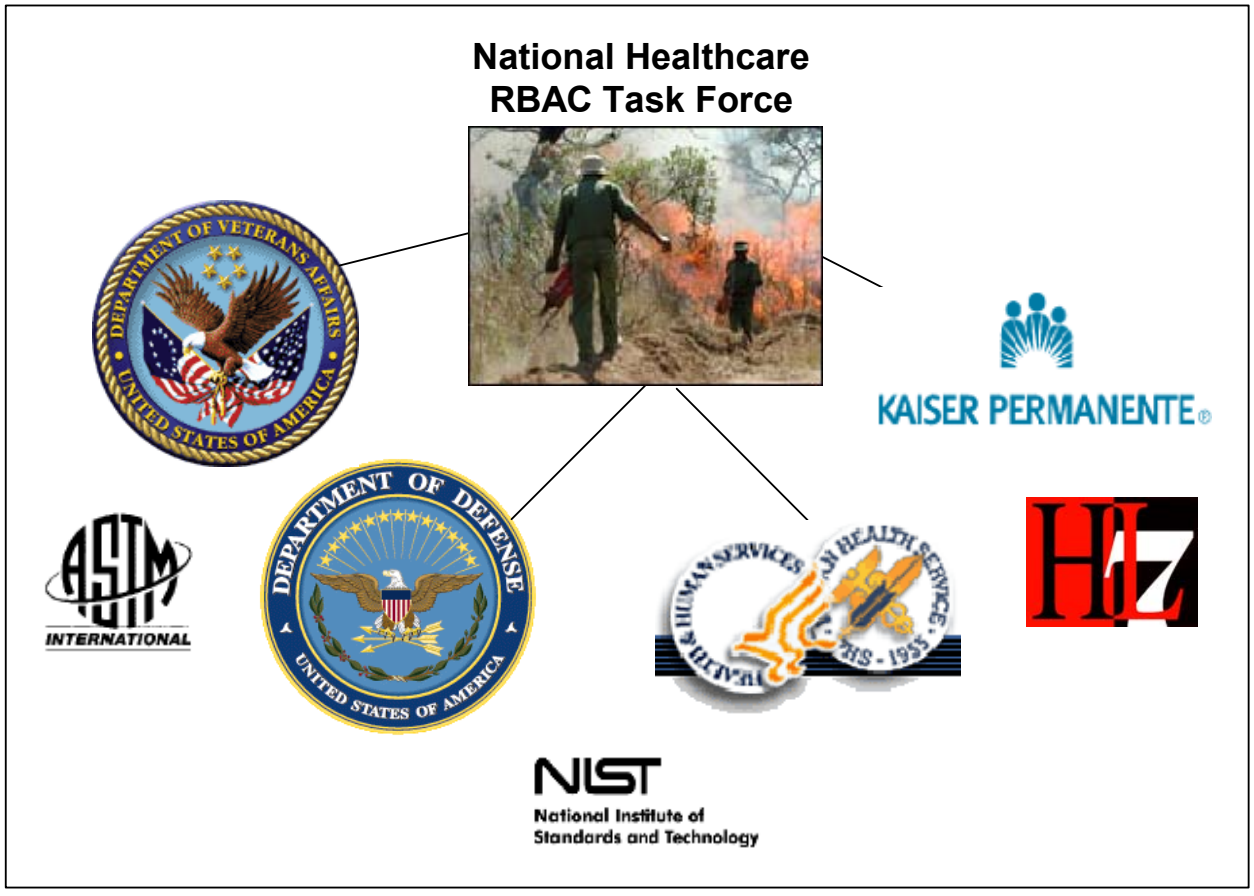

Figure 1: Healthcare RBAC Task Force

3.1 Healthcare RBAC Task Force

The Healthcare RBAC TF will collect and consolidate all information defined by the Enterprise RBAC Task Forces for development of a set of standard roles for the healthcare community. The Healthcare RBAC TF will present this information for use as a recommended standard to an SDO. The SDO would then turn the information from the Healthcare RBAC TF into a proposed RBAC standard for use within the healthcare community. The Healthcare RBAC TF acts as a liaison between the SDO and the Enterprise RBAC TFs.

Development of the roles themselves is outside of the scope of the Task Force activities.

3.2 Enterprise RBAC Task Force

Each Enterprise RBAC TF will use the role engineering approach as defined in this document to define tasks and permissions. Specific healthcare functional areas will be assigned to each Enterprise RBAC TF. When a specific functional area is completed, the list of tasks and permissions will be turned over to the Healthcare RBAC TF for further development.

As with the Healthcare RBAC TF, development of the roles themselves is outside of the scope of the Enterprise RBAC TF activities.

4 RBAC Terminology

4.1 Definitions

Table 1 is a list of terms that has been adopted for use within this document. The definition and source of each term are also listed.

Table 1: Definitions

Term	Definition	Source
User	A *user* is defined as a human being, but can be extended to include machines, networks, or intelligent autonomous agents.	[ANSI-RBAC]
Actor	An *actor* is defined as a healthcare worker involved in a step within a scenario. The actor and associated step are labeled in a sequence diagram.	VA RBAC TF
Role	A *role* is a job function within the context of an organization with some associated semantics regarding the authority and responsibility conferred on the user assigned to the role.	[ANSI-RBAC]
Basic Role	*Basic roles*, also called static roles, can be viewed as a precursor role that gives a person access to a "session" or “connection”.	ASTM
Functional Role	*Functional roles* reflect the essential business functions that need to be performed. *Functional roles* are defined by a set of standard healthcare tasks (e.g. Neurologist).	[Neumann/Strembeck]
Organizational Role	*Organizational roles* correspond to the hierarchical organization in a company in terms of internal structures. For a listing of healthcare organizational roles See ASTM E 1986-98 (e.g. Attending Physician).	[Neumann/Strembeck]
Role Group	A *role group* is a type of healthcare personnel warranting differing levels of access control.	Adapted from [ASTM 1986]
Junior Role	A *junior role* is when, in a role hierarchy, Role A is junior to Role B if Role B inherits all the permissions associated with Role A.	[XACML RBAC]
Senior Role	A *senior role* is when, in a role hierarchy, Role A is senior to Role B if Role A inherits all the permissions associated with Role B	[XACML RBAC]
Permission	*Permission* is an approval to perform an operation on one or more RBAC protected objects.	[ANSI-RBAC]

Term	Definition	Source
Operation	An *operation* is an executable image of a program, which upon invocation executes some function for the user. Within a file system, *operations* might include read, write, and execute. Within a database management system, *operations* might include insert, delete, append, and update. An *operation* is also known as a privilege.	[ANSI-RBAC]
Object	An *object* is an entity that contains or receives information. The *objects* can represent information containers (e.g., files or directories in an operating system, and/or columns, rows, tables, and views within a database management system) or *objects* can represent exhaustible system resources, such as printers, disk space, and CPU cycles. The set of *objects* covered by RBAC includes all of the objects listed in the permissions that are assigned to roles.	[ANSI-RBAC]
Task	A *task* is a collection of one or more scenarios.	[Neumann/Strembeck]
Work Profile	A *work profile* is a processing event that consists of all tasks performed by a user.	[Neumann/Strembeck]
Scenario	A *scenario* is an example of system usage in the form of action and event sequences. *Scenarios* are recorded as UML sequence diagrams.	[Neumann/Strembeck]
Storyboard	A *storyboard* is an HL7 healthcare scenario. See *Scenario*.	[HL7]
Step	A *step* is an action or event within a scenario.	[Neumann/Strembeck]
Session Roles	A *session role* is the role activated by a user session.	[ANSI-RBAC]
Action	An *action* is an operation on a resource.	[XACML]
Access	*Access* is performing an action.	[XACML]
Access Control	*Access control* is controlling access in accordance with a policy.	[XACML]
Policy	A *policy* is a set of rules, an identifier for the rule-combining algorithm and (optionally) a set of obligations. A policy may be a component of a policy set.	[XACML]
Resource	A *resource* can be data, service or system component.	[XACML]
Rule	A rule is the most elementary unit of a policy. A rule has a target, an effect and a condition.	[XACML]
Subject	A *subject* is an actor whose attributes may be referenced by a predicate.	[XACML]

4.2 Term Harmonization

Table 2 illustrates relationships among the ANSI Role Based Access Control (RBAC) standard (previously called the NIST RBAC Standard), Neumann/Strembeck, the XACML 1.0 standard, the XACML RBAC Profile, and the RBAC TF terminology. Wherever possible the ANSI RBAC term is preferred. RBAC TF terms harmonize ANSI RBAC and Neumann/Strembeck semantics into a single composite set.

Table 2: Term Harmonization

ANSI RBAC	Neumann/ Strembeck	XACML/XACML RBAC	RBAC TF
User	NA	NA	NA
NA	NA	Subject	Actor
Role	Functional Role	Attribute/Role	Functional Role
Role	Organizational Role	NA	Role Group
Permission	Permission	Rule/Permission	Permission
Operation	Operation	NA/Action	Operation
Object	Object	Resource	Object
NA	Work Profile	NA	Work Profile
NA	Task	NA	Task
NA	Scenario	NA	Scenario
NA	Step	NA	Step
Session Roles	NA	NA	NA

5 Role Engineering Models

The RBAC TF will use role engineering models to assist it in carrying out its activities. Models illustrate relationships between components of an abstract role system to the components of the role engineering process. The discussion below defines the various RBAC TF role engineering models.

5.1 Functional Roles and Role Groups

There are two types of high-level healthcare role models: Functional Roles and Role Groups.

Functional Roles consist of all the permissions on health information system objects needed to perform a task. Functional role names associate groups of permissions for convenience in assigning to users. A user may be assigned one or more functional roles, and thereby be assigned all of the permissions associated with a corresponding healthcare workflow. Permissions will ultimately be used to set the system operations (create, read, update, delete, execute, etc.) for data and software applications. *Functional roles may be found as entries in a user attribute certificate or stored in a distributed authorization directory.*

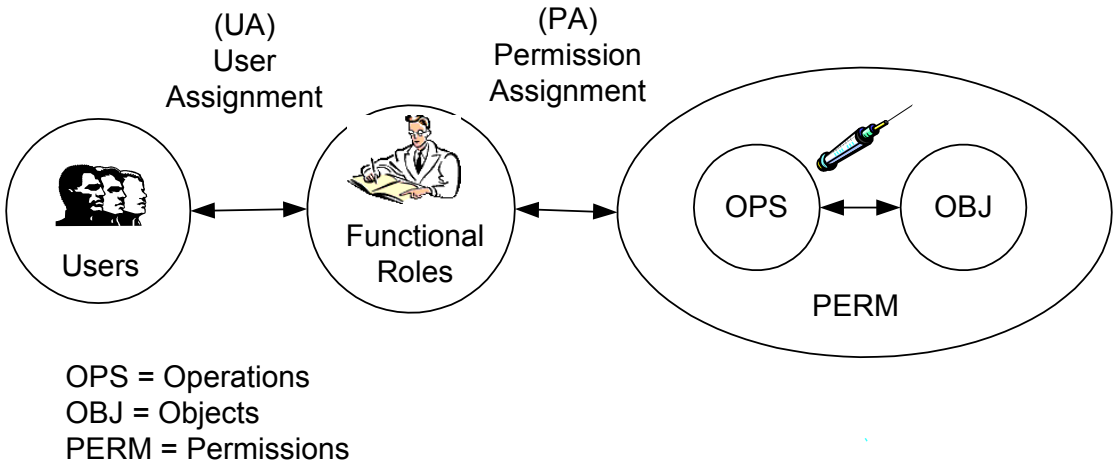

Figure 2: Role Structure (Adapted from ANSI RBAC Core RBAC Model)

Figure 2 illustrates the Functional Role, Permission, and Operation and Object relationships. This figure is an adaptation of the ANSI RBAC Core RBAC reference model. It does not include the concept of sessions/session roles, which is not part of the RBAC TF role engineering process.

ANSI-RBAC describes two other models: Hierarchical RBAC and Constrained RBAC. The role engineering process described herein recognizes these models but does not plan to use them.

Role groups place people in the organizational hierarchy as belonging to categories of healthcare personnel warranting differing levels of access control.[1] Similar to organizational roles, role groups allow users to participate in the organization's workflow (e.g., tasks) by job, title, or position but do not specify detailed permissions on specific information objects. Role groups allow a user to "connect" to a resource but do not grant authorizations. Some role group examples include: Physician, Pharmacist, Registered Nurse Supervisor, and Ward Clerk. *Role groups may be found as non-critical certificate extensions entries to an X.509 certificate as specified in ASTM 2212-00.*

As depicted in Figure 3, role groups define what specific healthcare tasks users are allowed to perform while functional roles define what authorizations are needed by an entity to access protected health information system (information technology) resources.

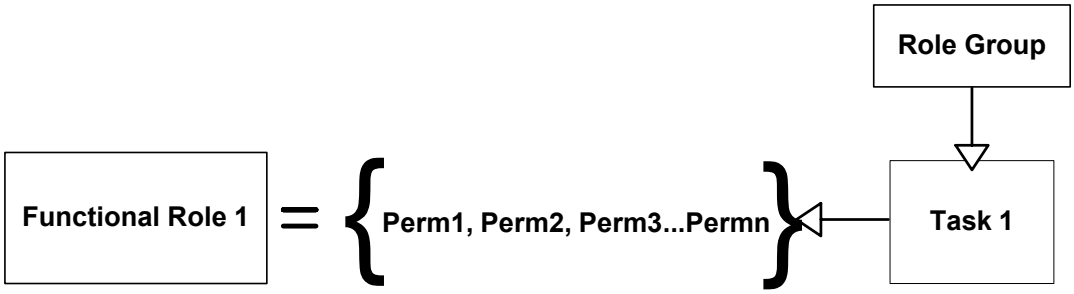

Figure 3: Functional Roles

Permissions are defined by operations (create, read, update, delete, and execute) on specific underlying health system information resources (objects). Some permissions include: Creating entries in laboratory results tables, certifying (signing) laboratory results entries, creating entries in patient orders tables, and creating, reading or updating patient allergy information.

The fundamental objective of the role engineering approach is to create a set of standard re-useable permissions. In this way, the Healthcare RBAC TF will create a set of common building blocks capable of defining an arbitrary number of functional roles to meet the needs of any work profile. Functional roles are standard to the extent that they incorporate standard permissions.

Note that there is no direct relationship between role groups and functional roles.

[1] See ASTM E1986-98 for a listing of healthcare personnel that warrant differing levels of access control.

5.2 Work Profiles, Tasks, Scenarios and Steps

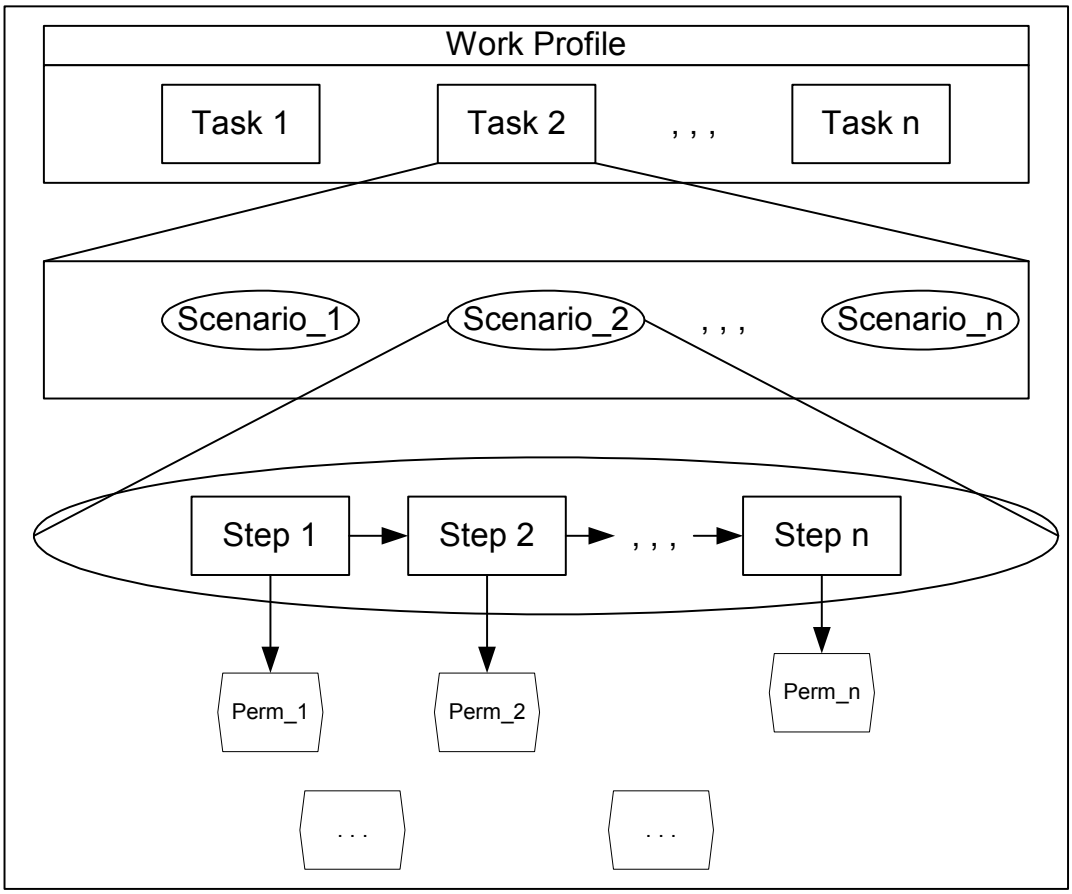

Figure 4: Scenario Model [Neumann/Strembeck]

The Scenario Model, as shown in Figure 4, illustrates the hierarchy of work profile, task, scenario, and step. Permissions are defined relative to steps (described in the role engineering process to follow).

In the scenario-based role-engineering approach each action and event within a scenario can be seen as a step that is associated with a particular access operation. Scenarios, which are applied in a particular order to reach a predefined task goal, act as sources for the derivation of permissions. The user performing a scenario must own all permissions that are needed to complete every step of the scenario.

5.3 Future Implementation of Standard Permissions

5.3.1 HL7 Reference Information Model (RIM)

The Healthcare RBAC TF adopts the HL7 Reference Information Model as the reference object model against which to define standard permissions. By adopting a standard

object model, the work of the task group is greatly simplified and need not consider implementation details of each collaborating member health information system.

Each enterprise would be responsible for mapping the model to its own systems and for implementing standard permissions as specific instances of operations (privileges) on the enterprise information objects. This approach allows enterprises significant flexibility in implementing the standard permissions than would otherwise be possible.

Figure 5 shows the HL7 method for decomposing RIM objects into a specification.

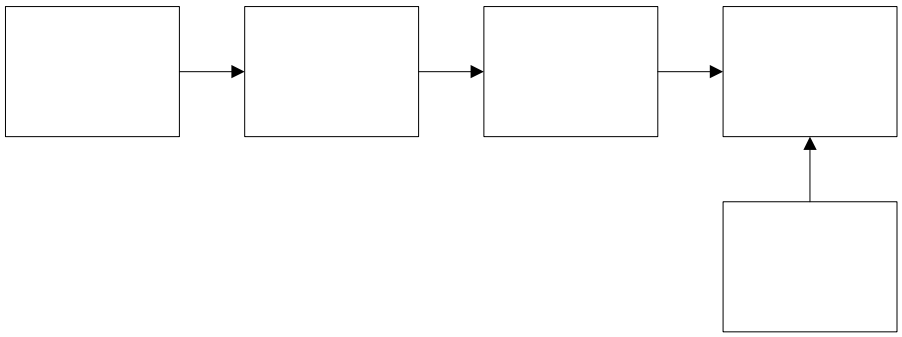

Figure 5: HL7 Method

5.3.2 eXtensible Access Control Markup Language (XACML)

In order to accomplish the goals of the RBAC TF effort, each enterprise must map the reference model to its own systems and, for implementing standard permissions as specific instances of operations, on the enterprise information objects. If this mapping is done in a site or application-specific way it may result in highly specific maintenance to encode and maintain each application. The eXtensible Access Control Markup Language (XACML) provides to developers the means to interface applications to a standard language for mapping standard permissions to operations on enterprise information objects. XACML is a markup language, based on an XML schema, used to express and evaluate access decisions on a system resource.

The XACML information exchange consists of an access request followed by an access decision. The application receiving the access decision then enforces that decision in a manner appropriate to the domain-specific context.

The request context contains information on the subject requesting the access, the resource being requested and the action requested. These three elements are known as a request target. Each XACML policy contains information on the subject, resource and action relevant to the policy (called the policy target). The policy, or policy set, relevant to the access request is retrieved and the access decision is made by comparing the request target to the policy target.

RIM F

The relevant policy contains one or more rules that determine the outcome of the request. Each rule contains information on the subject, resource and action relevant to the rule (called the rule target). The rules in the policy are then matched with the information in the request target. The access decision is determined by combining the results of rule evaluations.

A more detailed explanation of the XACML information exchange is defined by the OASIS eXtensible Access Control Markup Language Version 1.0, which became an OASIS standard on February 6, 2003.

In addition, XACML allows the enterprise to store the mapping information in a single repository. The use of an XACML repository will remove the burden of maintaining mapping information in several locations and possibly in several application-specific formats. Instead, mappings standard permissions used in role-based access control can maintained centrally using a single format.

More detailed specifications for RBAC using XACML are provided in [XACML RBAC]. There, a profile is written in XACML to define such constructs as role, permission, senior role, and junior role. However, this profile is intended for implementation and thus does not have a direct bearing on the role engineering process.

5.4 RBAC TF Role Engineering Model

Figure 6 illustrates the core unified RBAC TF Role Engineering Model. It consolidates the permission→work profile relationship, the core RBAC Model, and the HL7 RIM.

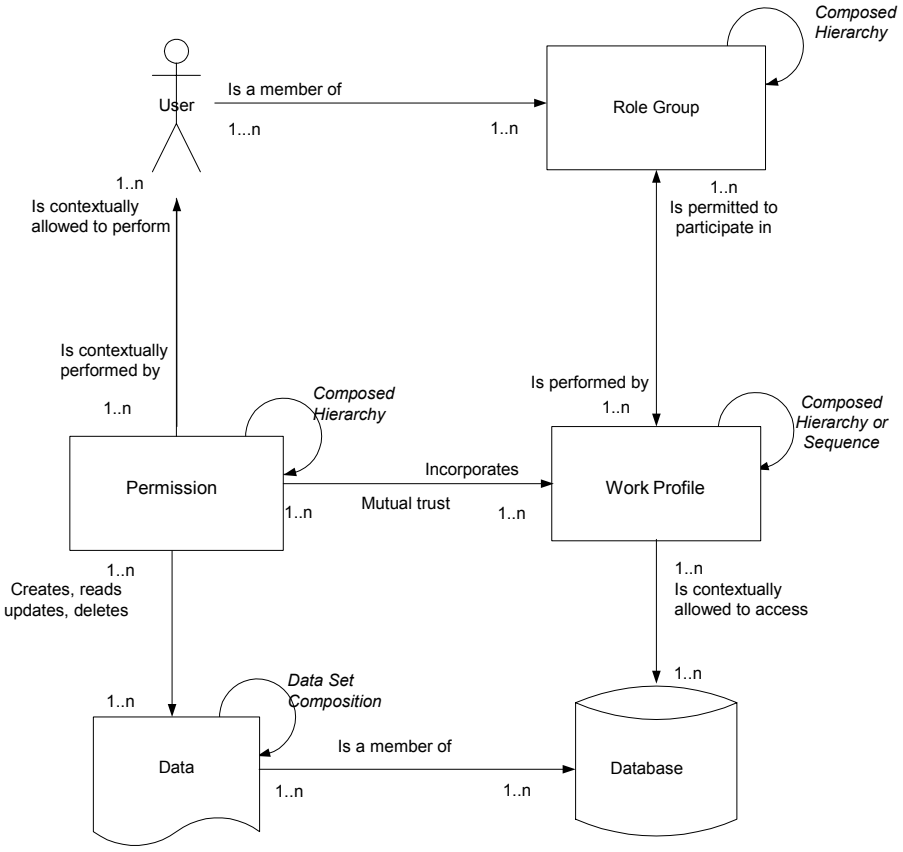

Figure 6: RBAC TF Role Engineering Model (Courtesy Siemens Medical Solutions)

Users are members of role groups permitted to participate in work profiles that contextually allow access to specific enterprise databases. From the user point of view, he has been granted the permissions (according to the principle of least privilege) that allow performing operations (create, read, update, delete, and execute) on protected information objects associated with the work profile scenarios.

The RBAC TF role engineering process creates abstract healthcare work profiles, permissions, and information model components (see Figure 7 grayed boxes). The RBAC TF abstract information model uses definitions and content from the HL7 RIM.

Using the model, RBAC TF members identify or create work profiles or tasks (e.g., HL7 storyboards=scenarios, etc.) to derive abstract permissions. These permissions further define operations on HL7 information objects.

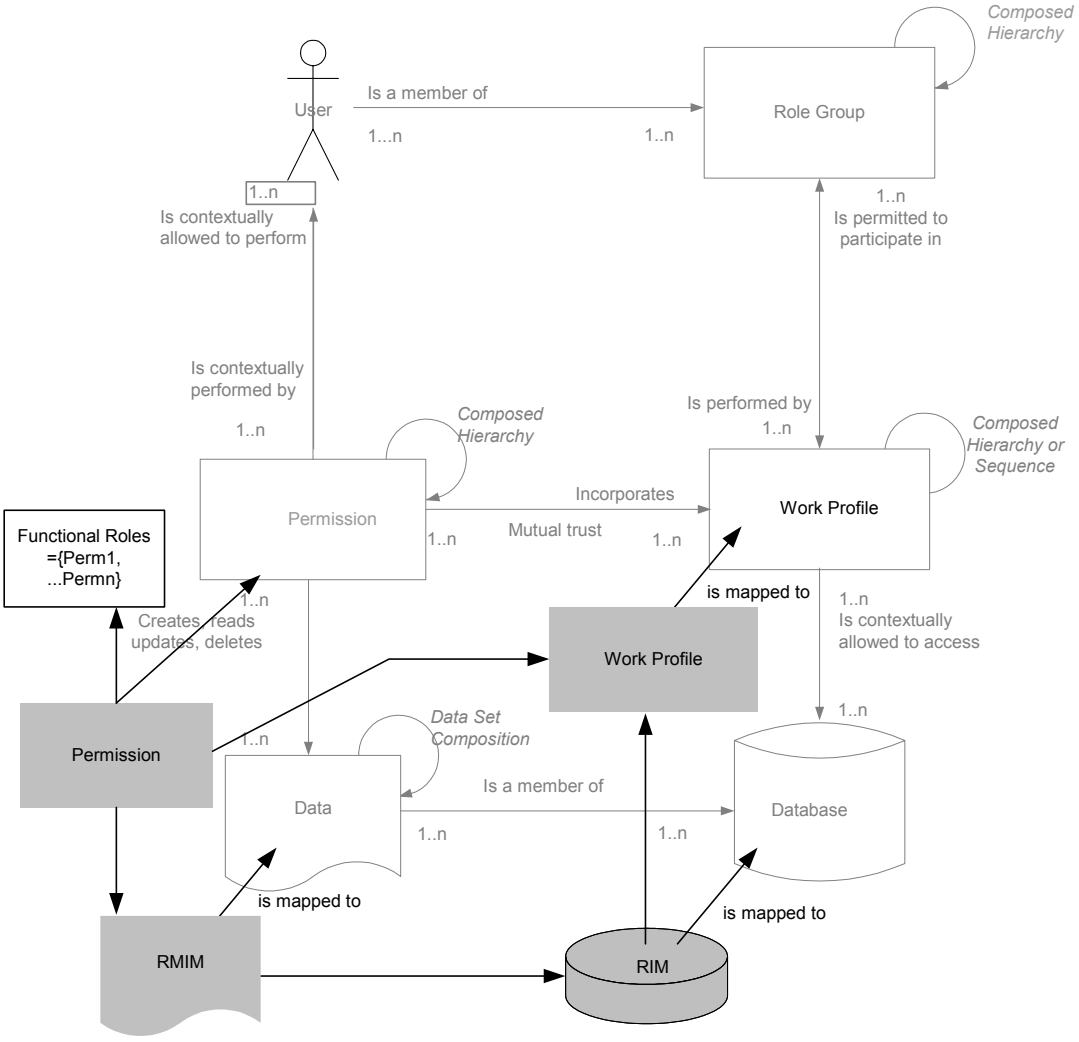

Figure 7: RBAC TF Role Engineering Model Overlay

In Figure 7, the RBAC TF role-engineering model has been overlaid upon the Figure 6 enterprise model. The white boxes represent the relationships between role components in an enterprise health information system. The gray boxes represent the RBAC TF model components.

6 Role Engineering Process

The fundamental objective of the role engineering approach is to create a set of standard re-useable permissions.

6.1 Role Engineering Process

[Neumann/Strembeck] provides a basis for defining roles using scenarios. Within this context, the following clarifications are made:

- Tasks reflect an organization's job functions and can be used to deduce permissions.
- The set of all work profiles a user is permitted to participate in reflects that user's operational roles.
- Role groups determine a user's authorization to connect to protected resources.
- Permissions determine what operations a user is permitted on health information system protected resources.
- Permissions may be used to define functional roles.
- Standard functional roles composed of standard permissions are defined to support inter-domain data transfer.
- Standard permissions can be mapped to specific health information system operations and protected health information
- Users are assigned to functional roles according to the principle of least privilege.

6.2 High-level View of the Role-engineering Process

The RBAC TF role-engineering process is based upon the scenario-driven [Neumann/Strembeck] process. [Neumann/Strembeck] is further clarified in this section for use within the RBAC TF. This Role Engineering Process replaces [Neumann/Strembeck] where applicable and as indicated. Where an apparent conflict exists, this document shall be authoritative.

6.2.1 Sources of Data

The sources of data and its use in the process are shown in Figure 8. It is expected that existing artifacts will be able to provide valuable input to the role engineering process. For example, the HL7 RIM contains candidate permissions. Where existing artifacts are considered inadequate for the purpose of the TFs, additional permission data can be discovered and recorded from such other sources as system access patterns. Data of this type can be elicited from system documentation or induced through interaction with system users.

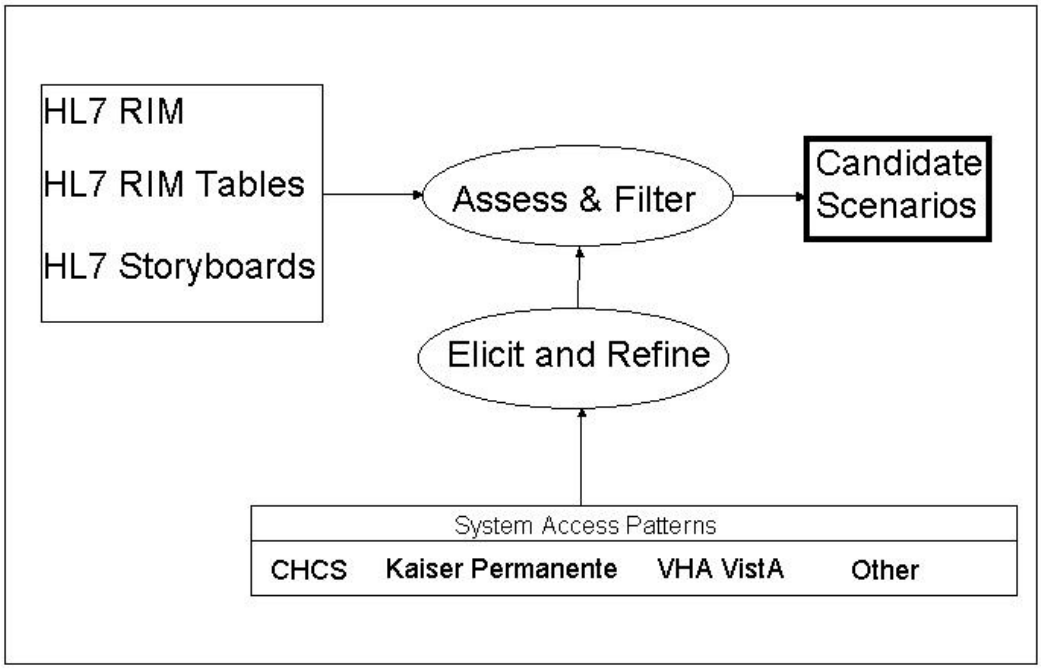

Figure 8: Sources of Data

6.2.2 Model Interrelations

Figure 9 illustrates how the Scenario Model is related to the remaining components of the TF role engineering process. Developing the Usage Scenarios, i.e., the Scenario Model, is the first step in the process.[2]

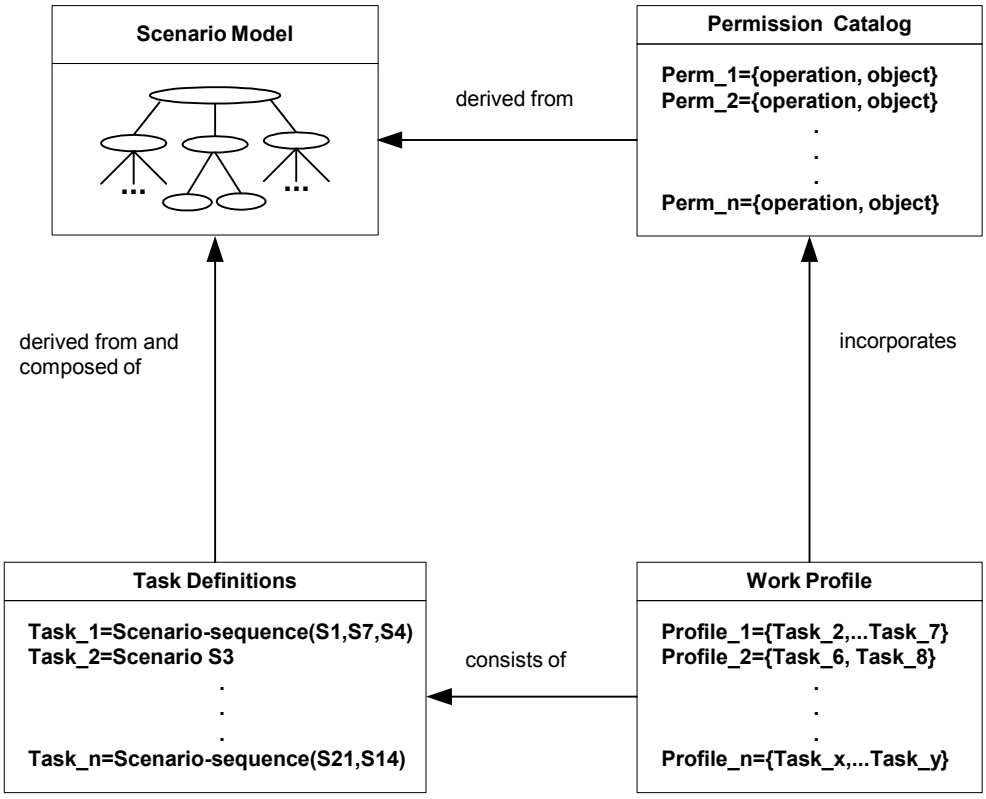

Figure 9: Interrelations of Scenario Model and Documents (Adapted from [Neumann/Strembeck])

[2] Establishing functional Work Groups prior to developing usage scenarios will provide a mechanism for grouping and categorizing scenarios.

6.2.3 RBAC Role Types

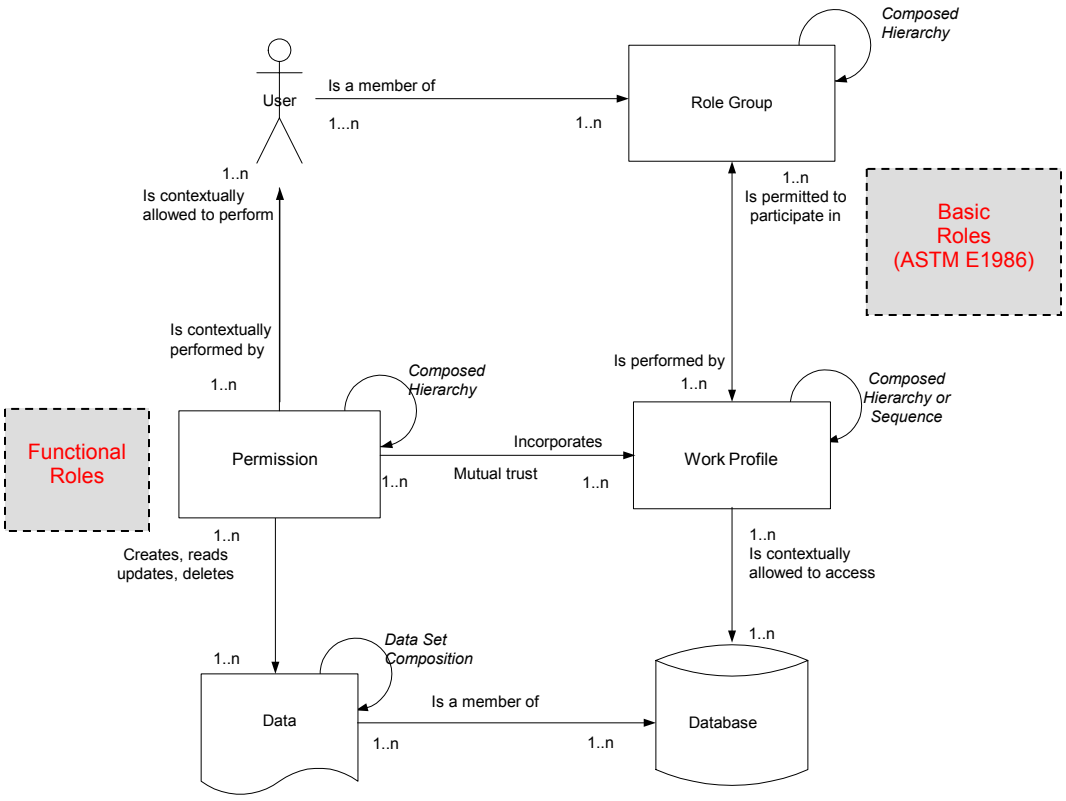

Figure 10. Basic and Functional Role Relationship

Figure 10 shows a relation between Basic roles and Functional roles through tasks and permissions. Basic roles place people within an organization's personnel (not necessarily organizational) structure into categories of personnel warranting differing levels of access control. Basic roles allow users to participate in the organization's workflow (e.g., tasks) by job, title, or position but do not specify detailed permissions on specific information objects. As stated earlier, Basic roles can allow a user to "connect" to a resource but do not necessarily grant finer-grain authorizations on protected information objects.

Functional roles contain the permissions that a user has available once the session is established and the roles are activated. Implementations of such roles are typically managed in applications, directories, and attribute certificates. While Basic roles are allowed to participate in workflows (decomposed into tasks and into scenarios etc, which a specific user performs), Functional roles specifically define, in terms of permissions, what authorizations are needed by an entity to access protected information technology or application resources.

6.2.4 Healthcare Scenario Roadmap

To better scope the magnitude of the role engineering effort, it is useful to identify the roles and the work tasks for each respective role. Figure 11 illustrates a TF-developed mapping that contains a list of high level scenarios mapped to ASTM E1986 basic healthcare roles. Licensed and non-licensed basic healthcare roles were derived from ASTM E1986, the Standard Guide for Information Access Privileges to Health Information. The list of healthcare system work tasks is derived from documentation, system access patterns, user interaction, and input from RBAC TF healthcare professionals and domain experts.

The cells containing an 'x' identify which roles are generally authorized to perform each work task. Further, each of the cells containing an 'x' will ultimately be contained within a scenario and have permissions derived using this RBAC role engineering process.

Legend: **x = authorized to perform** **o = not authorized to perform** **? = unknown** **red text = new content** **black text = TF approved content** **ASTM E1986 Basic Roles**	***Order Entry***	New/Change/Discontinue Laboratory Order	New/Change/Discontinue Radiology Order	New/Renew/Change/Discontinue/Refill Outpatient Rx Order	New/Change/Discontinue Inpatient Medication Order	New/Change/Discontinue Diet Order	New/Change/Discontinue Consult Order	New/Change/Discontinue Nursing Order	New/Change/Discontinue Standing Order(s) PRN	New/Change/Discontinue Verbal and Telephone Order	New/Change/Discontinue Supply Orders (ostomy, diabetic)	New/Change/Discontinue Prosthetic Order (wheelchair, crutches)	Sign Order(s)	Co-Sign Order(s)	***Review Documentation***	Review Patient Testing Reports	Review Chief Complaint	Review Medical History	Review Existing Orders	Review Patient Medications	Review Prescription Costing Information	Review Vital Signs/Patient Measurements	Review Identification and Lookup	Review Patient or Disease-Specific Clinical Guidelines	Review Alerts	Review Current Directory of Provider Information
Licensed Health Care Providers																										
Physician (MD/Allopath, Osteopath, Chiropractic, Naturopath, Homeopath)		x	x	x	x	x	x	x	x	x	x	x	x	x		x	x	x	x	x	x	x	x	x	x	x
Advanced Practice Registered Nurse (NP, NM, CAN, CNS)		x	x	x	x	x	x	x	x	x	x	x	x	x		x	x	x	x	x	x	x	x	x	x	x
Physician Assistant (PA)		x	x	x	x	x	x	x	x	x	x	x	x	o		x	x	x	x	x	x	x	x	x	x	x
Midwives		x	x	?	?	x	x	x	x	x	x	x	?	?		x	x	x	x	x	x	x	x	x	x	x
Registered Nurse (RN)		x	x	x	x	x	x	x	x	x	x	x	x	o		x	x	x	x	x	x	x	x	x	x	x
Pharmacist (DP)		x	o	x	x	o	o	o	o	x	x	o	o	o		x	x	x	x	x	x	x	x	x	x	x
Licensed Vocational Nurse (LVN)		x	x	o	o	x	x	?	x	?	o	o	o	o		x	x	x	x	x	x	x	x	x	x	x

Figure 11: Healthcare Scenario Roadmap Example

6.3 Initial Assumptions

The following initial assumptions are made regarding the activities of the RBAC TF.

- Scenarios are documented as UML sequence diagrams[3]
- Defining the RBAC model means the identification of standard permissions.
- Explicitly naming functional roles based upon standard permissions is not desired.
- All permissions will be defined in relation to HL7 RIM Version 3.0.

6.4 Detailed Process Description

This section is taken from and clarifies [Neumann/Strembeck]. It is keyed to corresponding sections of [Neumann/Strembeck] Section 4.

6.4.1 Identify and Model Usage Scenarios [Neumann/Strembeck 4.1]

6.4.1.1 Preliminary

Initial scenarios will be suggested from participating RBAC TF collaborators. The collaboration will initially seek out HL7 storyboards as the primary source for scenarios. Where storyboards are incomplete or not available, model scenarios will be derived from VHA, DoD, IHS and KP system-specific documentation or elicited from users.

Scenarios are the key to a systematic engineering approach leading to the development of permissions on protected health information objects. The RBAC TF will determine an initial set of model scenario names and appropriate methodology to document the scenario steps. Initial names may be arbitrary, but subsequently the names may be re-defined based on the semantics of the scenario content.

6.4.1.2 Detailed Process

In this sub-process sensible usage scenarios for the system are identified. At first the identified usage scenarios are described with a short sentence. A simple example is "Create a new patient record" in a hospital information system. Since these scenarios subsequently serve as the basis for the derivation of permissions and the definition of task and work profiles, it is essential that the step sequence within each scenario is explicitly defined and written down. Therefore each scenario is described by detailed structured text and a corresponding diagram. To identify scenarios and the corresponding step sequences, it is necessary to rely on the assistance of health-care domain experts like physician, nurse, and hospital clerk. In the final step of the scenario modeling sub-process each scenario is provided with a unique name to identify the scenario and to facilitate search operations within the scenario model.

STEP 1 ➡ Gather an initial list of Healthcare scenarios using HL7 storyboards and actual system access patterns.[4]

[3] To assist in automated processing, the sequence diagrams may be represented by state tables.
[4] See Attachment A for a list of functional areas suitable for scenarios.

STEP 2 ➡Assign each scenario a name using the RBAC TF nomenclature. For each scenario create structured text (steps) and a sequence diagram.
STEP 3 ➡ Validate and complete scenarios with input from healthcare domain experts.
STEP 4 ➡ Record consolidated list of scenarios. This is the RBAC TF scenario model.

6.4.2 Permission Derivation from Scenarios [Neumann/Strembeck 4.2]

6.4.2.1 Preliminary

To be valid, permissions must exhibit specific attributes as follows:

- Provide access rights to protected health information,
- Provide access rights to (clinical systems) patient demographic information,
- Must be to security- or confidentiality-relevant events
- Auditable
- Contains at least one operation and one object.

6.4.2.2 Detailed Process

Sub-process activities include identifying permissions necessary to system scenarios and populating the permission catalogue. Permission identification includes reviewing each scenario step and deciding which operation a user needs to be performed to complete this step. For each of these operations, a record containing an {operation, object} pair in the permission catalogue is defined. Figure 12 depicts a model permission catalogue.

STEP 1 ➡ Review scenario and identify operations needed to perform the step.
STEP 2 ➡ Find the associated object in the associated HL7 RMIM/DMIM Classes.
STEP 3 ➡ For each scenario step, record the associated (operation, object) pair in the permission catalogue.

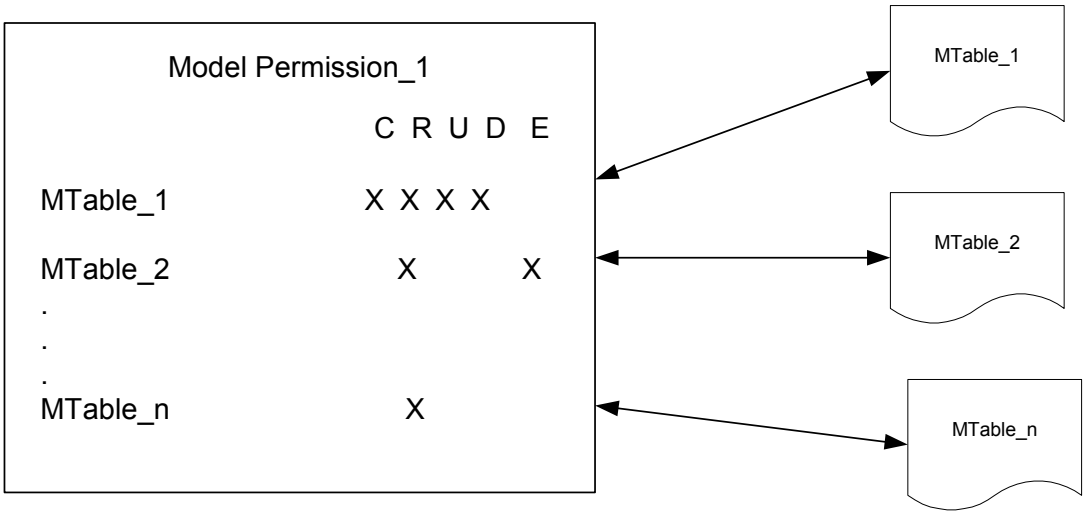

Figure 12: Model Permission Catalogue

6.4.3 Identification of Permission Constraints [Neumann/Strembeck 4.3]

Constraints are restrictions that are enforced upon access permissions. They can include contextual properties such as separation of duties, time-dependency, mutual exclusivity, cardinality or location, and are distinguished from healthcare policies that limit access to sensitive data. For instance, access to adoption information, Human Immunodeficiency Virus (HIV) test results and treatment, mental health visits, etc., may warrant specific permissions as opposed to constraints. Constraints may be enterprise specific.

For example, a constraint would occur for a permission when its definition is tied to cardinality. One example could be a permission that only one role performs at any given time. In that case, the constraint on the permission would be the cardinality specification.[5] Examples of permission constraints include:

- Head Nurse on a hospital floor (cardinality of 1)
- Chief of Staff (cardinality of 1)
- Lab Technician vs. Lab Technician Supervisor (separation of duties)
- Provider's access to a remote hospital which is not his/her primary workplace(location)
- Physician working in a clinic (time-dependency) vs. physician working in the ER (no time-dependency)

It is important that constraints be defined as a limitation on the specific permission and not simply a domain business rule. A domain business rule can be directly tied to the business process and thus the business rules are enforced via the application, not a permission constraint. Business rule logic should be handled via application logic rather than by the RBAC infrastructure. For example, the assumption that the "Read Progress Note" permission applies only to a signed progress note is a business rule and not a constraint. The "Read Progress Note" permission would, by definition, allow read access to the object that contains the progress note. The permission would not be expected to be conditioned by whether or not the progress note is signed. Examples of business rules include:

- Read signed progress note. An unsigned progress note is not considered complete.
- Within the VHA, dietitians have ordering privileges for vitamins, supplements, fiber, and nutritional liquids as 'Nutritional Orders' which are processed and filled through the facility's pharmacy department. These types of orders do not require a physician's prescription and are considered over-the-counter items. In the private sector and other healthcare institutions, these ordering activities would be accomplished by the dietitian placing a phone call to the physician and the physician, in turn, writing an order.

[5] This constraint could also be modeled as a constraint among roles. However, it is a best practice to place constraints on permissions when feasible. Thus, the restricted permissions could only be assigned to one role each, to preserve the desired condition. If it became necessary to include the constraint on multiple roles, it would then be necessary to place the constraint on the roles themselves.

- Within the VHA, both Residents and Attending Fellows require counter-signature of History and Physicals (H&Ps) and are not allowed to author any H&Ps. This is not a nationally accepted practice because of any licensure issues. Rather, within the VHA, this practice occurs for billing/finance purposes, thereby classifying this practice as a business rule.

6.4.4 Scenario Model Refinement [Neumann/Strembeck 4.4]

In this part of the process, the initial scenario model is reviewed and further refined. In essence one can distinguish two essential activities in this sub-process:

Concretion. Each step within each scenario is reviewed to see if it is complex enough to be described in more detail through its own sub-scenario.

STEP 1 ➡For each complex scenario, define sub-scenarios, as necessary.
STEP 2 ➡Update the scenario model.

Generalization. First, the current scenario model is reviewed to see if some similar scenarios exist. Second for each scenario additional (similar) scenarios are defined. Third, for each group of similar scenarios, determine if an abstract type for the scenarios can be defined. The scenarios are then grouped and a common abstract type is derived.

STEP 1 ➡ Search the scenario model for similar scenarios.
STEP 2 ➡ Consolidate the list of similar scenarios, eliminating duplicates.
STEP 3 ➡ Define an abstract type for the scenario (if necessary).
STEP 4 ➡ Group the similar scenarios and derive a common abstract type.

6.4.5 Definition of Tasks and Work Profiles [Neumann/Strembeck 4.5]

6.4.5.1 Preliminary

The RBAC task force is concerned with defining standard model permissions. To accomplish this goal, scenarios must be developed and (optionally) grouped into tasks for analysis. This process does not presume that scenarios must first be determined and grouped into tasks. In fact, the opposite may occur; tasks may be identified and broken into scenarios.

6.4.5.2 Detailed Process

In this sub-process, scenarios that logically belong together are combined into tasks. These tasks are then used to define work profiles.

- A task is a collection of scenarios that can be combined to perform a complex operation.

- A work profile consists of 1 or more tasks. Therefore, each work profile is a job description for a certain position within the organization under consideration.

The specifications for task and work profiles are defined with assistance from health-care domain experts. The most challenging part is to select the correct group of scenarios for a particular task.

STEP 1 ➡ Identify scenarios that logically belonged together.
STEP 2 ➡ Group the scenarios into tasks.

6.4.6 Derivation of a Preliminary Role-hierarchy [Neumann/Strembeck 4.6]

This section is excluded from the initial work of the RBAC TF. Further work on deriving role-hierarchies is left to the participating SDOs (i.e. HL7 and ASTM).

6.4.7 RBAC Model Definition [Neumann/Strembeck 4.7]

This section is excluded from the initial work of the RBAC TF and will instead be accomplished by the participating SDOs (i.e. HL7 and ASTM).

7 Applied Example

The first major activity within the RBAC role engineering process is to identify and model the usage scenario. Since the scenarios serve as the basis in which permissions, tasks and work profiles are defined, scenarios need to be explicitly defined and written. [Neumann/Strembeck]

The following sample storyboard, "Lab Frequency Order with Results", was obtained from an HL7 Orders/Observations Technical Committee. The storyboard depicts an Emergency Room Physician who evaluates a patient with complaints of chest pains and orders frequency laboratory tests whose results provide confirmation of the admitting diagnosis.

Portions of the sample storyboard will be applied to the RBAC Role Engineering Process, in which sequence diagrams with structures text (steps) will be created to represent the activity. Ultimately, role permissions will be derived and applied to HL7 DMIM objects.

7.1 Identify and Model Usage Scenarios (Reference Section 6.4.1)

STEP 1 ➡ Gather an initial list of Healthcare scenarios using HL7 storyboards and actual system access patterns.

Purpose: The purpose of this storyboard is to illustrate the order and result messaging related to lab frequency orders that report both preliminary and final results.

<|>**Presentation**</|>
Dr. Eric Emergency, an emergency room physician, sees a 45-year old male patient Adam Everyman, for chest pains. Myocardial infarction is suspected and the patient is admitted.

<|>**Activate Order**</|>
To determine whether the Adam Everyman has had a heart attack, Dr. Emergency orders a CPK with MB fractionation battery to be collected immediately and then every 8 hours for the next 2 days. The order is sent from the ordering system to the laboratory system.

<|>**Intent to Perform Order**</|>
A phlebotomist, Boris Bleeder, from the laboratory arrives to collect the first specimen shortly after the order is entered. Boris labels the specimen with labels printed on the STAT printer in the laboratory and then transports the labeled specimen back to the lab. Once the specimen arrives at the specimen processing section of the laboratory, lab tech Bill Beaker spins the tube of blood down in a centrifuge and delivers an aliquot of serum to the appropriate workstation. The laboratory system notifies the ordering system that the specimen has been received and that it intends to perform the requested series of tests.

<|>**Notify Laboratory Results**</|>
The total CPK test is performed and the result is transmitted from the laboratory system to the results reporting system. Since the MB fractionation will not be performed until the next run of isoenzymes, the partially resulted battery is reported as preliminary. Two hours later, the MB fractionation test is performed. The MB result is entered, the battery is marked as final and the results are sent to the results reporting system.

<|>**Intent to Perform Occurrence**</|>
At the next designated time 8 hours after the first specimen was obtained, the laboratory arrives to collect the next specimen in the series. Once the specimen arrives in the laboratory it is processed, delivered to the workstation and a notice is sent to the ordering system that the specimen has been received and that the lab intends to perform the requested tests on the current specimen.

<|>**Notify Laboratory Results**</|>
The total CPK test is performed and the result is transmitted from the laboratory system to the results reporting system. Since the MB fractionation will not be performed until the next run of isoenzymes, the partially resulted battery is reported as preliminary. After the MB fractionation test is performed, the battery is marked as final and the results are sent to the results reporting system.

The <|>**Intent to Perform Occurrence**</|> and the <|>**Notify Laboratory Results**</|> repeat for each requested specimen collection. Based on the series of results, Dr. Emergency concludes that his preliminary diagnosis of myocardial infarction was correct. Since the MB fraction peaks approximately 12-24 hours after heart attack, Dr. Emergency presumes Adam Everyman most likely infarcted very close to the time of admission and that no further myocardial damage occurred during this episode of care.

STEP 2 ➧Assign each scenario a name using the RBAC TF nomenclature. Create structured text (steps) and a sequence diagram for each scenario.

<u>Storyboard Name:</u> Frequency Lab Order with Results

For purposes of this example, two scenarios from the above "Frequency Lab Order with Results**"** complex storyboard, are depicted in the following Sequence Diagrams: "Intent to Perform Order – Collect Specimen", "Intent to Perform Order – Process Specimen" and "Intent to Perform Occurrence" in Figures 13-15. Each diagram in the scenario modeling sub-process is provided with a unique name to identify the scenario and to facilitate search operations within the scenario model. [Neumann/Strembeck]

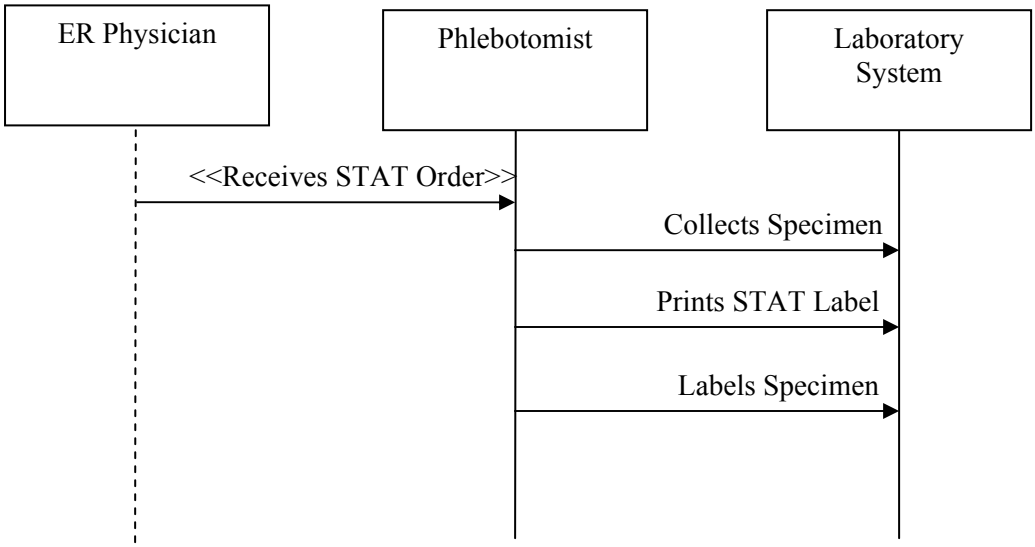

Figure 13: Intent to Perform Order – Collect Specimen Scenario

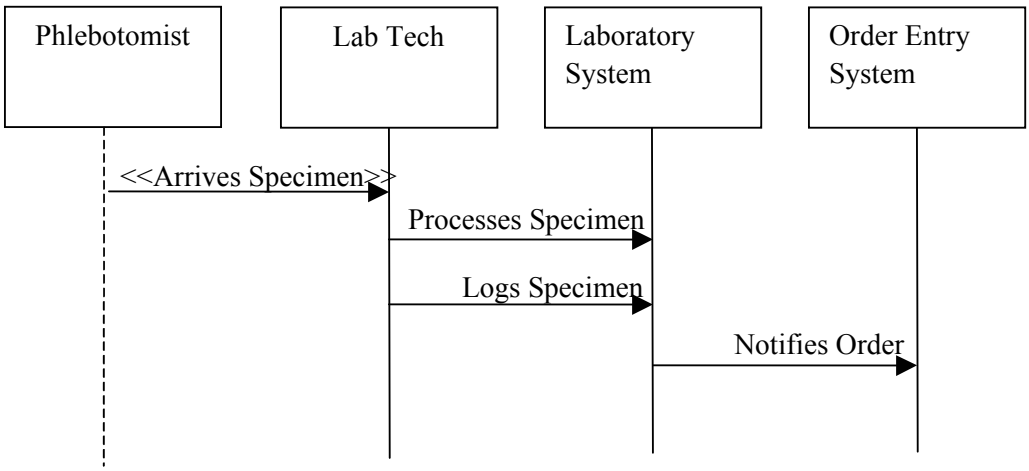

Figure 14: Intent to Perform Order – Process Specimen Scenario

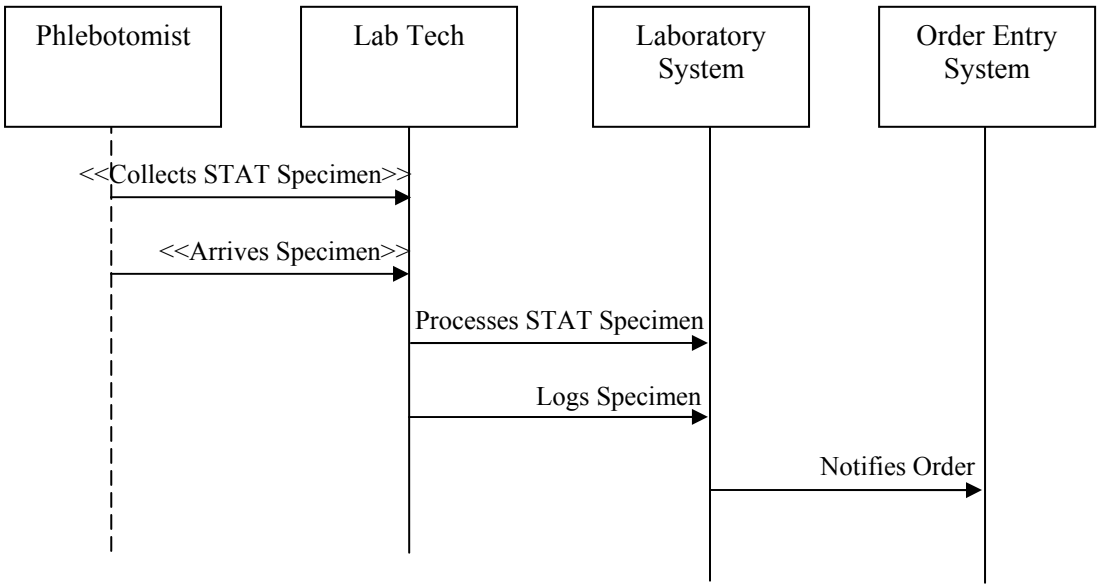

Figure 15: Intent to Perform Occurrence Scenario

STEP 3 ➡ Validate and complete scenarios with input from healthcare domain experts.

At this point, the scenarios would be reviewed and validated by other domain experts within the TF.

STEP 4 ➡ Record consolidated list of scenarios. This is the RBAC TF scenario model.

The storyboard or workflow name "Frequency Lab Order with Results" and its sub-scenarios are recorded as shown in the following Table 3. (New items in tables will be identified using italicized text.)

Table 3: Scenario Recordation

Workflow	Scenario
Frequency Lab Order with Results	*Intent to Perform Order - Collect Specimen*
Frequency Lab Order with Results	*Intent to Perform Order - Process Specimen*
Frequency Lab Order with Results	*Intent to Perform Occurrence*

7.2 Permission Derivation from Scenarios (Reference Section 6.4.2)

The second major activity in the process is to derive permissions that correspond with the step-sequence. The operation that a subject (e.g. user) performs in order to complete a step is identified and stored as {operation, object} pairs in the permission catalogue. [Neumann/Strembeck]

STEP 1 ➡ Review scenario and identify the actor who performs each step.

In this part of the process, each scenario is reviewed and the actors and steps are identified. Table 4 contains the actor-to-step mapping.

Table 4: Identification of Actors and Steps

Workflow	Scenario	Actor	Step
Frequency Lab Order with Results	Intent to Perform Order - Collect Specimen	*Phlebotomist*	*Receives STAT Order*
Frequency Lab Order with Results	Intent to Perform Order - Collect Specimen	*Phlebotomist*	*Collects Specimen*
Frequency Lab Order with Results	Intent to Perform Order - Collect Specimen	*Phlebotomist*	*Prints STAT Label*
Frequency Lab Order with Results	Intent to Perform Order - Collect Specimen	*Phlebotomist*	*Labels Specimen*
Frequency Lab Order with Results	Intent to Perform Order - Collect Specimen	*Phlebotomist*	*Sends Specimen*
Frequency Lab Order with Results	Intent to Perform Order - Process Specimen	*Phlebotomist*	*Arrives Specimen*
Frequency Lab Order with Results	Intent to Perform Order - Process Specimen	*Lab Tech*	*Processes Specimen*
Frequency Lab Order with Results	Intent to Perform Order - Process Specimen	*Lab Tech*	*Logs Specimen*
Frequency Lab Order with Results	Intent to Perform Order - Process Specimen	*Laboratory System (Lab Tech, Phlebotomist and/or Pathologist?)*	*Notifies Order*
Frequency Lab Order with Results	Intent to Perform Occurrence	*Phlebotomist*	*Collects STAT Specimen*
Frequency Lab Order with Results	Intent to Perform Occurrence	*Phlebotomist*	*Arrives Specimen*
Frequency Lab	Intent to Perform	*Lab Tech*	*Processes STAT Specimen*

Workflow	Scenario	Actor	Step
Order with Results	Occurrence		
Frequency Lab Order with Results	Intent to Perform Occurrence	*Lab Tech*	*Logs Specimen*
Frequency Lab Order with Results	Intent to Perform Occurrence	*Laboratory System (Lab Tech, Phlebotomist and/or Pathologist?)*	*Notifies Order*

STEP 2 ➡ Find the associated operation and object in the associated HL7 DMIM Classes.

In the review of the HL7 DMIM classes, identify the objects associated with the steps from the scenarios. Table 5 contains the step-to-operation and object mapping. The system operations are defined as "C, R, U, D, E", or create, read, update, delete, and execute, respectively.

Table 5: Identification of Objects

Workflow	Scenario	Actor	Step	Operation	HL7 DMIM Object
Frequency Lab Order with Results	Intent to Perform Order - Collect Specimen	Phlebotomist	Receives STAT Order	*R, C*	*Order, Observation*
Frequency Lab Order with Results	Intent to Perform Order - Collect Specimen	Phlebotomist	Collects Specimen	*U, U, R*	*Observation, Order, WorkList*
Frequency Lab Order with Results	Intent to Perform Order - Collect Specimen	Phlebotomist	Prints STAT Label	*C*	*Device*
Frequency Lab Order with Results	Intent to Perform Order - Collect Specimen	Phlebotomist	Labels Specimen	*C*	*Container*
Frequency Lab Order with Results	Intent to Perform Order - Process Specimen	Phlebotomist	Collects STAT Specimen	*U, U*	*Observation, Order*
Frequency Lab Order with Results	Intent to Perform Order - Process Specimen	Phlebotomist	Arrives Specimen	*U, U*	*Observation, Order*
Frequency Lab Order with Results	Intent to Perform Order - Process Specimen	Lab Tech	Processes STAT Specimen	*U*	*Observation*
Frequency Lab Order	Intent to Perform Order -	Lab Tech	Logs Specimen	*U*	*Observation*

Workflow	Scenario	Actor	Step	Operation	HL7 DMIM Object
with Results	Process Specimen				
Frequency Lab Order with Results	Intent to Perform Order - Process Specimen	Laboratory System (Lab Tech, Phlebotomist and/or Pathologist?)	Notify Order	*U*	*Order*
Frequency Lab Order with Results	Intent to Perform Occurrence	Phlebotomist	Collects STAT Specimen	*C, U, R*	*Observation, Order, WorkList*
Frequency Lab Order with Results	Intent to Perform Occurrence	Phlebotomist	Arrives Specimen	*U, U*	*Observation, Order*
Frequency Lab Order with Results	Intent to Perform Occurrence	Lab Tech	Processes STAT Specimen	*U*	*Observation*
Frequency Lab Order with Results	Intent to Perform Occurrence	Lab Tech	Logs Specimen	*U*	*Observation*
Frequency Lab Order with Results	Intent to Perform Occurrence	Laboratory System (Lab Tech, Phlebotomist and/or Pathologist?)	Notifies Order	*U*	*Procedure*

STEP 3 ➡ For each scenario's step, combine the associated {operation, object} pair in the permission catalogue.

In this step, the operations and objects from the last table are merged into an associated pair. The associated pairs for this scenario are listed in Table 6.

Table 6: Identification of Associated {Operation, Object} Pairs

Workflow	Scenario	Actor	Step	{Operation, Object}
Frequency Lab Order with Results	Intent to Perform Order - Collect Specimen	Phlebotomist	Receives STAT Order	*{R, Order}, {C, Observation}*
Frequency Lab Order with Results	Intent to Perform Order - Collect Specimen	Phlebotomist	Collects Specimen	*{C, Observation}, {U, Order}, {R, WorkList}*
Frequency Lab Order with Results	Intent to Perform Order - Collect Specimen	Phlebotomist	Prints STAT Label	*{C, Device}*
Frequency Lab Order with Results	Intent to Perform Order - Collect Specimen	Phlebotomist	Labels Specimen	*{C, Container}*
Frequency Lab Order with Results	Intent to Perform Order - Process Specimen	Phlebotomist	Collects STAT Specimen	*{U, Observation}, {U, Order}*
Frequency Lab Order with Results	Intent to Perform Order - Process Specimen	Phlebotomist	Arrives Specimen	*{U, Observation}, {U, Order}*
Frequency Lab Order with Results	Intent to Perform Order - Process Specimen	Lab Tech	Processes STAT Specimen	*{U, Observation}*
Frequency Lab Order with Results	Intent to Perform Order - Process Specimen	Lab Tech	Logs Specimen	*{U, Observation}*
Frequency Lab Order with Results	Intent to Perform Order - Process Specimen	Laboratory System (Lab Tech, Phlebotomist and/or Pathologist?)	Notify Order	*{U, Order}*
Frequency Lab Order with Results	Intent to Perform Occurrence	Phlebotomist	Collects STAT Specimen	*{C, Observation}, {U, Order}, {R, WorkList}*
Frequency Lab Order with Results	Intent to Perform Occurrence	Phlebotomist	Arrives Specimen	*{U, Observation}, {U, Order}*
Frequency Lab Order with Results	Intent to Perform Occurrence	Lab Tech	Processes STAT Specimen	*{U, Observation}*
Frequency Lab Order with Results	Intent to Perform Occurrence	Lab Tech	Logs Specimen	*{U, Observation}*

Workflow	Scenario	Actor	Step	{Operation, Object}
Frequency Lab Order with Results	Intent to Perform Occurrence	Laboratory System (Lab Tech, Phlebotomist and/or Pathologist?)	Notifies Order	*{U, Procedure}*

Basic steps, such as "Notify Order", will likely be included in many different scenarios. These steps will be normalized as shown in the next section. Each permission is registered only once in the Permission Catalogue.

7.3 Identification of Permission Constraints (Reference Section 6.4.3)

Permission Constraints are identified and documented. Such constraints are restrictions that are enforced upon access permissions and may often be enterprise-specific. Constraints will be documented within this role engineering process describing the associated actor and permission constraint (i.e. location restriction, time limitations) using a free-form text field.

7.4 Scenario Model Refinement (Reference Section 6.4.4)

The Scenario Model Refinement process activity involves reviewing the initial scenario model to ensure that it contains complexity details (Concretion). The scenario is then compared against other similar scenarios to possibly define an abstract type (Generalization). [Neumann/Strembeck]

Concretion:

STEP 1 ➡For each complex storyboard, define multiple scenarios as necessary.

The storyboard used for this example was already deemed complex and decomposed into multiple scenarios in Section 7.1.

STEP 2 ➡Update the scenario model.

The scenario model for this storyboard was updated in Section 7.1.

Generalization:

Our example storyboard represents the ordering, collecting, processing and resulting of frequency STAT laboratory orders. For generalization purposes, an example of a similar storyboard for comparison and possibly abstract definition could include one-time laboratory orders, non-panel laboratory orders, and laboratory tests to be collected and resulted with ASAP, Routine, Pre-Op, etc. timing.

STEP 1 ➡ Search the scenario model for similar scenarios.

In this step, search the complete list of associated pairs for duplicates. Duplicates are color coded in Table 7. (Note: Rows without colors are not duplicates.)

Table 7: Associated Pairs – Duplicates

Scenario	Actor	Step	{Operation, Object}
Intent to Perform Order - Collect Specimen	Phlebotomist	Receives STAT Order	{R, Order}, (C, Observation}

Scenario	Actor	Step	{Operation, Object}
Intent to Perform Order - Collect Specimen	Phlebotomist	Collects Specimen	(U, Observation}, {U, Order}, (R, WorkList}
Intent to Perform Order - Collect Specimen	Phlebotomist	Prints STAT Label	{C, Device}
Intent to Perform Order - Collect Specimen	Phlebotomist	Labels Specimen	{C, Container}
Intent to Perform Order - Process Specimen	Phlebotomist	Arrives Specimen	{U, Observation}, {U, Order}
Intent to Perform Order - Process Specimen	Lab Tech	Processes Specimen	{U, Observation}
Intent to Perform Order - Process Specimen	Lab Tech	Logs Specimen	{U, Observation}
Intent to Perform Order – Process Specimen	Laboratory System (Lab Tech, Phlebotomist and/or Pathologist?)	Notifies Order	{U, Observation}, {U, Order}
Intent to Perform Occurrence	Phlebotomist	Collects STAT Specimen	{U, Observation}, {U, Order}, {R, WorkList}
Intent to Perform Occurrence	Phlebotomist	Arrives Specimen	{U, Observation}, {U, Order}
Intent to Perform Occurrence	Lab Tech	Processes STAT Specimen	{U, Observation}
Intent to Perform Occurrence	Lab Tech	Logs Specimen	{U, Observation}
Intent to Perform Occurrence	Laboratory System (Lab Tech, Phlebotomist and/or Pathologist?)	Notifies Order	{U, Observation}, {U, Order}

STEP 2 ➡ Consolidate the list of similar Steps and {Operation, Object}, eliminating duplicates.

Normalize the list of actors, steps and {Operation, Object} by grouping similar steps and identical {Operation, Object} pairs. Duplicates will be eliminated, as shown in Table 8. Isolated pairs (no color) are recorded but not grouped.

Table 8: Associated Pairs – Normalized

Scenario	Actor	Step	{Operation, Object}
Intent to Perform Order - Collect Specimen	Phlebotomist	Receives STAT Order	*{R, Order}, (C, Observation}*
Intent to Perform Order - Collect Specimen	Phlebotomist	Collects Specimen	*(U, Observation}, {U, Order}, (R, WorkList}*
Intent to Perform Occurrence	Phlebotomist	Collects STAT Specimen	
Intent to Perform Order - Collect Specimen	Phlebotomist	Prints STAT Label	*{C, Device}*
Intent to Perform Order	Phlebotomist	Labels Specimen	*{C, Container}*

Scenario	Actor	Step	{Operation, Object}
- Collect Specimen			
Intent to Perform Order - Process Specimen	Phlebotomist	Arrives Specimen	*{U, Observation}, {U, Order}*
Intent to Perform Occurrence	Phlebotomist	Arrives Specimen	
Intent to Perform Order - Process Specimen	Lab Tech	Processes Specimen	*{U, Observation}*
Intent to Perform Occurrence	Lab Tech	Processes STAT Specimen	
Intent to Perform Order - Process Specimen	Lab Tech	Logs Specimen	*{U, Observation}*
Intent to Perform Occurrence	Lab Tech	Logs Specimen	
Intent to Perform Order – Process Specimen	Laboratory System (Lab Tech, Phlebotomist and/or Pathologist?)	Notifies Order	*{U, Observation}, (U, Order}*
Intent to Perform Occurrence	Laboratory System (Lab Tech, Phlebotomist and/or Pathologist?)	Notifies Order	

STEP 3 ➡ Define an abstract type for the scenario (if necessary).

Not applicable to example.

STEP 4 ➡ Group the similar scenarios and derive a common abstract type.

Steps are then labeled as 'permissions' and given unique permission identifications within a Permissions Catalogue. Table 9 contains the abstract and basic permissions derived from the set of scenario steps from Table 8 above and the associated {Operation, Object} pairs. The Unique Permission ID will soon be namespaced (i.e. OR_Adm might represent "Order, Admission Type") so that both the Role Group and the Permissions being performed are easily identifiable.

Table 9: Permission Catalogue

Unique Permission ID	Actor	Abstract Permission Name	Basic Permission Name
Perm_1	Phlebotomist	*Receives STAT Order*	{R, Order}, {C, Observation}
Perm_2	Phlebotomist	*Collects Specimen*	{U, Observation}, {U, Order}, {R, WorkList}
Perm_3	Phlebotomist	*Prints STAT Label*	{C, Device}
Perm_4	Phlebotomist	*Labels Specimen*	{C, Container}
Perm_5	Phlebotomist	*Arrives Specimen*	{U, Observation}, {U, Order}
Perm_6	Lab Tech	*Processes Specimen*	{U, Observation}
Perm_7	Lab Tech	*Logs Specimen*	{U, Observation}
Perm_8	Laboratory System (Lab Tech, Phlebotomist and/or Pathologist?)	*Notifies Order*	{U, Observation}, {U, Order}

7.5 Remaining Process Activities

The remaining RBAC TF process activities, Definition of Tasks and Work Profiles (reference section 6.4.5) and Derivation of a Preliminary Role-hierarchy (reference section 6.4.6), cannot be represented until more scenarios are defined and normalization of the collected data occurs. (Note: The RBAC Model Definition (reference section 6.4.7) is performed at the conclusion of the TF effort by the standards organization.)

8 References

[ASTM 1986] American Society for Testing and Materials. *ASTM Standard E1986-98: Standard Guide for Information Access Privileges to Health Information*, 1998.

[Collins] A. Collins, T. Cooper, MD, G. Martin and E. Powers. *Role-based Access Control: A Sensible Approach.* Healthcare Information and Management Systems Society, 2000.

[HL7] L. Dailey-Evans. *HL7 Lab Frequency Order Storyboard.* Health Level 7 Orders Technical Committee, 2002.

[IETF] Marshall, *Security Audit and Access Accountability Message Data Definitions for Healthcare Applications*, Internet Draft, Internet Engineering Task Force, December 2003.

[Neumann/Strembeck] G. Neumann and M. Strembeck. *A Scenario-driven Role Engineering Process for Functional RBAC Roles*, June 2002.

[ANSI-RBAC]. Information Technology Industry Council. *American National Standard for Information Technology - Role-Based Access Control*, ANSI INCITS 359-2004, , 2004.

[SAIC] SAIC Security Engineering. *Working Paper – Implementing A Role-Based Access Control Policy for CHCS II*, 1999.

[SAIC2] Staggs, David. *XACML in the VHA Development Environment Version 1.0*, Science Applications International Corporation Secure Business Solutions Group, March 2004.

[XACML]OASIS. *eXtensible Access Control Markup Language (XACML) Version 1.0.* OASIS Standard. 18 February 2003.

[XACML RBAC] OASIS. *XACML Profile for Role Based Access Control (RBAC) Committee Specification 01*, 13 February 2004.