Fall 9-1-2015

Analysis of Healthcare Workflows in Accordance with Access Control Policies

Sandeep Kumar Lakkaraju
Dakota State University

ANALYSIS OF HEALTHCARE WORKFLOWS IN ACCORDANCE WITH ACCESS CONTROL POLICIES

A graduate project submitted to Dakota State University in partial fulfillment of the requirements for the degree of

Doctor of Science
in
Information Systems

September, 2015

By
Sandeep Kumar Lakkaraju

Project Committee Chair:
Dr. Yong Wang

Project Committee
Dr. Ashley Podhradsky
Dr. Dianxiang Xu
Dr. Mark Hawkes

DAKOTA STATE
dsu
UNIVERSITY

DISSERTATION APPROVAL FORM

This dissertation is approved as a credible and independent investigation by a candidate for the Doctor of Science in Information Systems degree and is acceptable for meeting the dissertation requirements for this degree. Acceptance of this dissertation does not imply that the conclusions reached by the candidate are necessarily the conclusions of the major department or university.

Student Name: SANDEEP KUMAR LAKKARAJU

Dissertation Title: **ANALYSIS OF HEALTHCARE WORKFLOWS IN ACCORDANCE WITH ACCESS CONTROL POLICIES**

YWang

Dissertation Chair: ______________________ Date: 10/17/2015

Committee member: ______________________ Date: 10/16/2015

Committee member: ______________________ Date: 10/18/2015

Committee member: ______________________ Date: Oct. 19, 2015

Original to Office of Graduate Studies and Research

Acid-free copies with written reports to library

ACKNOWLEDGMENT

I would like to acknowledge the following for their assistance and motivation in successfully completing all the requirements for the D.Sc.IS program. Firstly I would like to thank my family members, Parents: L.V.R. Murthy and S. L. Parvathi, my wife Monica, my brother Santhosh Lakkaraju and his wife Sravanthi, for their never ending love and support on all the things that I wish to do with focus. I thank my project committee chair Dr. Yong Wang, for his unending support, for sharing his knowledge and spending time to discuss on various issues. I thank the committee members: Dr. Ashley Podhradsky, Dr. Dianxiang Xu, and Dr. Mark Hawkes who are always available and have encouraged me and provided support in completing the dissertation. I thank Dr. Dianxiang Xu personally, for his encouragement and his guidance in the initial stage of my dissertation and for his continued support till this date. It is Dr. Xu who has helped me gain an in depth knowledge about this topic. I thank Dean of Graduate studies and Research Dr. Omar El-Gayar for his guidance and extended support. I also would like to thank Dr. Ashley Podhradsky, with whom I have been working as a graduate assistant. Dr. Podhradsky was very supportive and encouraging. I would like to thank Dr. Surendra Sarnikar and Dr. Amit Deokar for helping me learn and understand IS research. Also I would like to thank Dr. Sreekanth Malladi, without whose initial push, I won't be able to learn the access control and other information security related issues. Last but not least, I would like to thank Paula Herron Jensen for her endless smiles and all of my friends who have extended their support in my good and hard times.

I will treasure all these precious moments I have shared with the faculty, staff, and friends especially my family at Dakota State University.

ABSTRACT

Healthcare information systems deal with sensitive data across complex workflows. They often allow various stakeholders from different environments to access data across organizational boundaries. This elevates the risk of exposing sensitive healthcare information to unauthorized personnel, leading 'controlling access to resources' a major concern. To prevent unwanted access to sensitive information, healthcare organizations need to adopt effective workflows and access control mechanisms. It is well-known that aligning security policies with business objectives is a challenging task. As of now, many healthcare organizations are mainly using role based access control. It is important for them to develop workflows and properly assign roles to tasks without the policies causing obstructions. Also many healthcare organizations are not yet considering or do not know how to accommodate the 'context' element as a crucial element in their workflows and access control policies. We envision the future of healthcare where 'context' will be considered as a crucial element. We can accommodate context through a new element 'environment' in workflows, and can accommodate context in policies through well-known attribute based access control mechanism (ABAC). As of now, it is hard to identify what policies are being applied on a particular workflow activity, and also it is hard to identify if all of the access policies are being used and which of those policies are not being used. This dissertation mainly addresses these problems by developing two methodologies one for each scenario: (i) analyzing workflow instances for obstructions due to static and dynamic authorization policies through a new algorithm that allows organizations to properly assign users to tasks without the policies causing obstructions, and (ii) identifying workflow activities that are

not being protected by access control policies and improving the workflow activities and/or existing access control policies through integrating workflow activities and attribute based access control policies using SARE (Subject, Action, Resource, and environment) elements.

DECLARATION

I hereby certify that this project constitutes my own product, that where the language of others is set forth, quotation marks so indicate, and that appropriate credit is given where I have used the language, ideas, expressions or writings of another.

I declare that the project describes original work that has not previously been presented for the award of any other degree of any institution.

Signed,

Sandeep Kumar Lakkaraju

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

The wide adoption of information technology across healthcare organizations has increased the need for developing efficient workflows. Literature has shown that healthcare organizations with efficient workflows have more success rate in adopting healthcare information technology (HIT) into their organizations (Sittig, Krall, Kaalaas-Sittig, & Ash, 2005). Effective care delivery is possible through workflows with competent activities (Campbell, Sittig, Ash, Guappone, & Dykstra, 2006). Nevertheless, security and privacy of healthcare information have become major challenges for healthcare organizations (Akinyele et al., 2011; Alhaqbani & Fidge, 2007). In particular, effective access control mechanisms are needed to protect sensitive information from being exposed to unauthorized personnel.

1.1 Workflows

Workflows are used to describe the pattern of tasks to be executed by users to achieve business objectives. According to Welch (2014), "Efficient clinical workflow saves time, saves money, and saves lives. And in today's industry, workflow can have a significant effect on reimbursement". A workflow can be implemented in many ways, possibly unboundedly many ways, called instances. An instance may involve more than one workflow activity. Workflows can be very complicated, especially in a complex environment like healthcare which may involve various subjects trying to perform actions on certain resources in multiple environments, thus requires controlling the access of resources by subjects. For a subject to perform an action on a resource in an environment, that subject should be authorized to perform the intended action. In this research, Business Process Modeling Notation (BPMN) has been used to develop workflows. BPMN (Business Process Modeling Notation) is visualization for business process workflows (Giaglis, 2001). The various elements of BPMN are given in Figure 1-1. We use four types of elements, defined as follows: Events that can be start or finish, distinguishable by size of circles; Activities (Tasks) have an “id” (A1; A2 etc.) and possibly user icons on the upper right corner of the rectangle, and groups indicate a group of activities; Gateways are diamond shaped, with multiple input channels; Conditional gateways are plain diamonds with only one output channel that is based on the evaluation of a condition; Parallel gateways have diamonds with a ‘+’ sign inside. They have multiple input/output channels indicating that control flows on those channels in parallel; Sequences (flows) link tasks together, and associations are used to associate activities with flow objects; some of the extra events used are also shown in the Figure 1-1 representing Binding of Duties (BOD) policy and Separation of Duty (SOD) policy, and a release event (each of which are explained in detail in chapter 3).

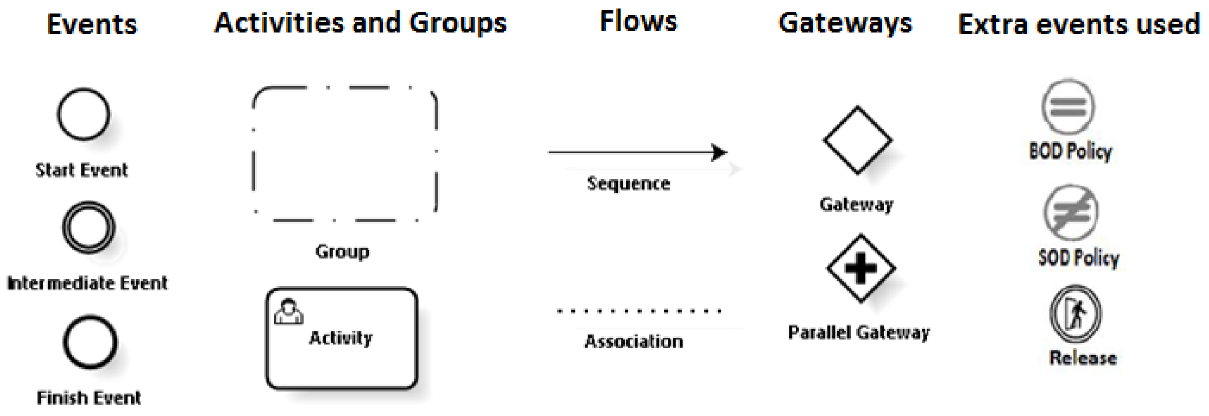


Figure 1-1 Workflow elements

A workflow involves execution of a series of certain activities which help in achieving a goal (Chaari, Biennier, Amar, & Favrel, 2004). These workflows have to be executed in a secure way, which is made possible through access control.

1.2 Access Control

Inter-organizational systems allow users to access and share data beyond organizational boundaries and therefore needs proper authorization mechanisms to protect sensitive information from being exposed to unauthorized personnel. An access control policy defines the conditions to which access to resources can be granted and to whom (Ferreira, Cruz-Correia, Antunes, & Chadwick, 2007). With the increasing complexity of information systems, access control methods have evolved from Mandatory Access Control (MAC), Discretionary Access Control (DAC), Task Based Access Control (TBAC), Context Based Access Control (CBAC), Role-Based Access Control (RBAC), to Attribute-Based Access Control (ABAC) (Xu & Zhang, 2014). Each of these access control mechanisms is discussed below in detail:

Mandatory Access Control (MAC)

MAC mechanism mainly categorizes the data by attaching a 'label'. Some of the examples can be: classified information, unclassified information, public, private, sensitive, secret, etc. The security clearance will depend on the assigned 'label'. MAC is implemented through rule based access control. Along with 'label', user should mention 'need to know' element. Disadvantages of MAC may include: If someone wants to make a change, they have to make changes to each and every rule and therefore it is hard to modify rules. It works best for group of users with similar needs. MAC cannot support organization wide access control model and therefore cannot support inter-organizational access control. MAC cannot support contextual information.

Discretionary Access Control (DAC)

According to Trusted Computer System Evaluation Criteria, DAC is a "means of restricting access to objects based on the identity of subjects and/or groups to which they belong. The controls are discretionary in the sense that a subject with certain access permission is capable of passing that permission (perhaps indirectly) on to any other subject" (USDoD, 1985). It is a centralized security model where access is distributed at the interest of the owner. It is assumed that every resource has an 'owner' and that owner will dictate the permissions to access and access control is maintained by only one department. It is ideal for small organizations, but it would be hard to maintain access control in large sized organizations. Disadvantages of DAC may include: As only one department maintains the access control policies, it would be time consuming to make changes in a large organization. DAC mainly concentrates on inter-organizational access control issues, but not intra-organizational access control issues. Security in DAC is discretionary. DAC does not consider contextual information.

Task Based Access Control (TBAC)

TBAC – 'Task' based access control mechanism is suitable for dynamic information processing activities, distributed computing, etc. It mainly concentrates on 'enterprise' level rather than system-centric level and the main disadvantage of TBAC is that it does not deal with contextual information.

Context Based Access Control (CBAC)

CBAC – is mainly a characteristic of 'firewall software', where it will filter the information based on pre-determined TCP and UDP packets. It helps in identifying and avoiding denial of services, and also issuing real time alerts. Disadvantage of CBAC is that it lacks user-aspect of the access control policies.

Role Based Access Control (RBAC)

RBAC is one of the popular access control mechanisms. A user is assigned to ‘roles’, where each role has certain ‘permissions’ to perform certain ‘operations’ on certain ‘objects’. It is more useful to attain separation of duties. Multiple users can be assigned to a single rule, and role permissions can be ‘inherited’ from other roles. RBAC is extremely capable of ‘Separation of duties’. Disadvantages of RBAC may include: Role based access control mechanism does not consider contextual information. RBAC is very popular and sound for fewer roles (around 10), but as the users in organization increases, so will the roles increase, leading to ‘role explosion’. Every time a user resigns/removed from the organization, he/she has to be removed from the role; failure to which may result in unwanted consequences. If user of Role ‘r’ at certain organization tries to access a resource at a different organization (in case of inter-organizational systems), that user should be placed on an access list of the other organization, making it a cumbersome task in large organizations.

Alhaqbani & Fidge (2007) has reviewed three main access control techniques (mandatory access control, discretionary access control, and role-based access control) through a case study and validated that when applying individually, these access control methodologies cannot fully secure the resources in a complex healthcare system, and therefore authors have suggested to use combination of all the three access control techniques. Authors have specified seven security and privacy requirements for a healthcare environment and later on showed how each of these three access control mechanisms cannot satisfy these seven security and privacy requirements, and therefore has suggested a combined access control protocol (Alhaqbani & Fidge, 2007)

ABAC has the core features of the previous access control mechanisms. It provides dynamic access control capability. ABAC does not depend on subject-resource relation, but on various attributes (of subject, resource, environment, etc.). ABAC accommodates MAC and DAC and TBAC by using rules and attributes as labels, RBAC through subject attributes, and CBAC through environment attributes. National Institute of Standards and Technology (NIST) indicates ABAC as a recommended access control language which can improve 'information sharing within organizations and between diverse and disparate organizations while maintaining control of that information' (Hu, Scarfone, & Kuhn, 2013).

Coming to attribute base access control (ABAC), it is "an access control method where subject requests to perform operations on objects are granted or denied based on assigned attributes of the subject, assigned attributes of the object, environmental conditions, and a set of policies that are specified in terms of those attributes and conditions" (Hu et al., 2013). ABAC is developed from RBAC. The preferability of ABAC over other access control mechanisms is demonstrated in detail in (Yuan & Tong, 2005). In ABAC, the authorization elements are defined in terms of attributes, rather than identities, of subjects, actions, resources, and environments (SARE). Attributes are characteristics of these entities that are predefined and pre-assigned by an authority. ABAC basically involves subject attributes, action attributes, resource attributes, and environment attributes. Subject attributes help in identifying a user (or an operation or a process) with its characteristics rather than its role; for example: job title, user name, age, etc. A resource attribute is a characteristic of a resource (record, data, etc.); for example: medical elements of patient record, non-medical elements of patient record, etc. Action attributes help in identifying the type of action the subject will perform on a resource; for example: read, write, etc. Environment attributes are operational or situational characteristics, such as current time and IP address. Consider the following access control rule: Any doctor can read and write into patient health record (PHR) only for his/her designated patients from internal network. The subject attributes include ‘doctor’, the action attribute is ‘read’ and ‘write’, and the resource attribute is ‘patient health record’, and the environment variable is ‘internal network’ and the condition is that the doctor can only access the health records of the patients designated to him/her.

1.3 HIPAA Compliance of Workflows and Access Control

Health Insurance Portability and Accountability Act is a federal law that was intended to protect the health information and health insurance coverage for working families. HIPAA was enacted on August 21, 1996. HIPAA protects the insurance coverage of individuals or working families in the cases of job loss and the standards are meant to improve the efficiency and effectiveness of the nation’s health care system encouraging the use of electronic data interchange in the United States healthcare system (HHS.Gov, 2003).

HIPAA mainly defines three rules (Siriwardena, 2013): (a) privacy rule, (b) transaction rule, and (c) security rule. The privacy rule mainly concentrates on revealing of confidential patient information, who can disclose what and to whom (HHS.Gov, 2003); The transaction rule concentrates on transaction processes of electronic health information which should follow certain regulated formats (e.g.: when certain claims are reported electronically). The security rule concentrates on the administrative process of creating, collecting, storing, maintaining and dealing with sensitive protected electronic health information (e.g.: access to health record in a hospital, etc.) (HHS.Gov, 2003). Based on these definitions, it is justified to say that this research is compliant with the privacy and security rules of the HIPAA.

Workflow activities and access control policies developed in this dissertation are developed to limit the revealing of confidential patient information to only authorized personnel, and efficient policies are developed to create and store protected health information. More about HIPAA is discussed below followed by a discussion on how HIPAA compliance is accommodated in this dissertation.

The healthcare providers undertake various measures and new responsibilities are added to safeguard patient information. The cheapest, quickest and most convenient way of providing the health information to the patients can be done with the help of the Internet electronically. The people involved in designing the web applications and hosting the websites have to work abiding the responsibilities and privacy rules stated by HIPAA. The Department of Health and Human Services summarizes that the HIPAA regulations are mainly recommended to: (a) give consumers control over their health information, (b) create boundaries around how medical records can be used and released, (c) ensure the security of personal health information, and (d) establish accountability for the use and release of medical records.

The HIPAA security rules demand that all the entities should safeguard administrative, physical and technical areas. Each section should define procedures which ensure that medical data is secured and protected (Cole, 2002; Leyva, 2014). The HIPAA privacy rule came into effect in 2003 stating that HIPAA not only provides patients access to their own medical records, but they also have the right to know who has accessed their health information over the preceding six years. If a healthcare provider fails inadequately providing either of the privacy rules, the patients now have a right to lodge complaints and force those in possession of this data to make it available to them. In this context the HITECH act came in to existence.

The Health Information Technology for Economic and Clinical Health Act, which is a part of American Recovery and Reinvestment Act of 2009, contains the incentives expanding the legislation of exchange of electronically protected health information (Leyva, 2014). HITECH Act also widens the scope of privacy and security protections available under the HIPAA Act. The HITECH Act creates a system of incentives to encourage practices to implement EHRs and disincentives to penalize slow adoption. This act includes the following improvised security requirements: (a) requirement to notify patients and HHS of PHI (Protected Health Information) security breaches, (b) new HIPAA regulations regarding business partners (PHRs, HIEs) and enforcement of penalties (c) restrictions on the sales and marketing of PHI, (d) ensuring that patients have access to their electronic health information, and (e) accounting of disclosures of PHI to patients. This act provides a right to individuals to obtain their PHI in an electronic format (i.e. ePHI), in the cases where the provider has implemented the EHR system. It is really important to understand the long-term needs of the patients and healthcare professionals which can be solved by implementing the EHR systems, leading to the evolution of HITECH Act. The privacy restrictions should be stricter and stringent information access should be provided to only concerned healthcare officials.

The Health Insurance Portability and Accountability Act will help in providing federal protection for personal health information held by covered entities and gives patients an array of rights with respect to that information. Mainly there are 2 sections in Health Insurance Portability and Accountability Act. They are: Title I and Title II. Title I covers the insurance of people who do not have permanent jobs. Title II concentrates on the information technology operations, which will affect the electronic data interchange. HIPAA is monitored by the HHS (U.S department of Health and Human Services). HHS administers the deadlines and requirements for organizations. Complying with time, Privacy rule and Security Rule are included in the HIPAA. These help in protecting the privacy of patient health information (PHI) and describing the best available options for the companies to maintain the integrity, and confidentiality of the ePHI (electronic protected health information). HIPAA affects all those health care providers, health plans, Medicare prescription drug card sponsors, etc. who deal with the creation, maintaining, transmitting of ePHI. HIPAA wants organizations to monitor the availability, integrity, and confidentiality of ePHIs, defend themselves from threats or hazards to ePHI, maintain detailed auditing of any kind of electronic transfer, prevent the loss of medical records through the removable devices, ensure that the organization complies with HIPAA rules, and maintain the security measures to protect ePHI. These guidelines are to be strictly followed by the health care organizations. Failing to follow these rules may lead to both monitory and imprisonment punishments.

According to Janssen (2014), Security Breach is defined as “a security breach is any incident that results in unauthorized access of data, applications, services, networks and/or devices by bypassing their underlying security mechanisms. A security breach occurs when an individual or an application illegitimately enters a private, confidential or unauthorized logical IT perimeter”. This primary kind of breach found in health care is the data breach. It occurs when some crucial information about the patients (like their last names, first names, social security numbers, addresses, and medical information) gets into wrong hands. There are many cases where a laptop with sensitive information is stolen and is misused for some illegal activities. Any illegal use of this kind of crucial information must be stopped. Guidelines provided by the HIPAA help the healthcare industry to become more efficient and more secure. They help in reducing the costs for healthcare payers, as well as in reducing the

Medicare and Medicaid-related fraud. It helps in providing extra security and less room for mistakes in transferring the PHI (protected health information). Many health care organizations started offering more and more security by building strong privacy, transaction and code, and security standards.

HIPAA can only be applied on Protected Health Information (PHI), and if there is no evidence of PHI, HIPAA cannot be applied. Whenever a set of PHI data is being created, edited, transferred, or involved, the person or organization that is dealing with the PHI must verify the issues such as: (i) access, (ii) protection, and (iii) transmission. (i) Access: Who can access the information, what can they access, who creates the usernames and passwords, how are the username and passwords created, etc. (ii) Protection: How is the data encrypted, where is the data stored, and how secure is the environment, what type of encryption methods are in use, etc. (iii) Transmission: Where is the data being transferred, how secure is the transfer process, what is the type of transfer mechanism being used (email, hard drives, laptops, other devices, etc.).

Implementing HIPAA compliance involves the steps below: (i) establishing a process to develop/improve policies (ii) administering security functions (iii) actively tracking personnel changes, and (iv) examining and recording the policy application (K. Cole, 2002).

The privacy rule involves: (i) Minimum necessary use: where an organization should delineate people who may request to have their hands on PHI. Organizations should also delineate the health information into various categories and should define the level of access an employee should have to access that particular information, and any other specific regulations that may apply; (ii) notice of privacy practice: Organizations should make understand the employees and any other stakeholders about their privacy practices, (iii) it also provides the patients with advanced rights of access about their health information. Upon patient's request, a healthcare entity should provide information sharing policies, and patients can request with whom their information may be shared or not shared.

The security rule involves confidentiality, integrity, and availability of information. Confidentiality involves restriction of access to information. Integrity imposes strict restrictions on type of modifications that can be performed on the protected health information. Availability ensures that recognized users have access to PHI whenever they require. This can be implemented in three ways as shown in this dissertation: developing policies in plain English, implementing through computer models (such as workflows and ABAC), and using technical mechanisms. In this dissertation, the access control policies and workflow activities are HIPAA compliant because:

(a) This methodology requires to develop and use uniform SARE standards

(b) workflow activities and access control policies use same SARE attributes

(c) access control policies consider environment variables to allow or deny access

(d) policies can be modified based on patient's interests

(e) The privacy will be enhanced by considering the environment attributes, which would further allow the administrator to know what information has been accessed from which location and by whom.

1.4 Addressing Obstruction Free Workflow Instances

It is not always possible to determine if a workflow instance has obstructions, with just casual inspection. Automated analysis is very much desirable, since it gives a high degree of confidence about the analysis. Literature concerning analysis of other similar infinite-state models such as cryptographic protocols is replete with security violations that could only be found with automatic and formal approaches (Lowe, 1996). This is the problem we consider in this paper. We designed an algorithm that answers the following question: 'Given a workflow W, an instance $\hat{w}$ of it, and a set of authorization policies E, is $\hat{w}$ obstruction-free with respect to E?' We can use this in turn to answer the question, 'If U is a set of users playing the roles R of tasks T in W, what are the set of substitutions of U to R that are obstruction-free?'

1.5 Integrating Access Control and Workflow

Many organizations treat access control and workflow as two separate issues. As both workflow specifications and access control polices can be complex, it is likely that there exists discrepancy between workflow activities and access control policies. For example, some workflow activities may not be adequately regulated by the access control policies. To address this issue, we propose a methodology for integrated modeling and analysis of workflows and ABAC policies.

The proposed methodology views workflow activities in terms of subject, resource, action, and environment. For example, we may use "Doctor writes a prescription within the intranet", rather than "Doctor writes a prescription", where "within the intranet" serves as an environment constraint of the activity. Specifying environment attributes in a workflow will provide more contextual information for analysis. It is also consistent with the ABAC concepts. Through cross-examination of inter-organizational workflow activities and access control policies, the proposed methodology can identify: (a) Whether there are workflow activities that are not covered by the access control policies, (b) whether there are unused access control policies, (c) how many policies are covering an activity, and (d) whether the workflow activities and the access control policies need to be modified or refined.

1.6 Contributions

Our contributions through this dissertation include addressing the below research questions: (A) ‘Given a workflow W, an instance $\hat{w}$ of it, and a set of authorization policies E, is $\hat{w}$ obstruction-free with respect to E?’ We can use this in turn to answer the question, ‘If U is a set of users playing the roles R of tasks T in W, what are the set of substitutions of U to R that are obstruction-free?’ and (B) proposing ‘environment’ as a crucial element in a workflow activity along with Subject, Action, and Resource, and (C) developing a methodology to integrate workflow activities and access control policies, and using this methodology to address the questions: (a) whether there are any workflow activities that are not covered by the access control policies, (b) whether there are any unused access control policies, (c) how many policies are covering each activity, and (d) whether the workflow activities and the access control policies need to be modified or refined.

The rest of the dissertation is divided into multiple chapters and sub-chapters. Chapter 2 includes the system design research methodology, which is design science research. Chapter 3 includes the analysis of workflows in business processes for obstructions due to authorization policies. Chapter 3 also includes an algorithm (one of the contributions), and evaluation of the algorithm in a healthcare scenario. Chapter 4 includes the methodology to integrate attribute based access control policies and workflows in healthcare organizations. Chapter 4 further includes assumptions in developing the methodology, demonstrating the approach, and expected results of the integrated model. Chapter 5 includes a case study where we will show an implementation and analysis of the integrated methodology in a healthcare scenario, where we take a healthcare workflow and a set of access control policies and test the model using WSO2 Identity Server, followed by future work, references, and appendices.

2 SYSTEM DESIGN (RESEARCH METHODOLOGY)

2.1 Preface

Chapter 1 focuses on introduction to what is a workflow, types of workflow elements, what is an access control, types of access control mechanisms, HIPAA compliance of workflows and access control mechanisms, addressing obstruction free workflow instances, and integrating access control and workflow. Following the introduction, the type of research methodology used in this dissertation is discussed here in chapter 2. Chapter 2 discusses the types of research methodologies design science vs behavioral science research, importance of design science research and why it is considered for this dissertation.

2.2 Designscience vs Behavioral science

Information System (IS) research is majorly influenced by two paradigms: design science paradigm, and behavioral science paradigm (Hevner, March, Park, & Ram, 2004). Hevner et al. (2004) indicates that both these paradigms are basic elements for information science where the design science research helps in creating new objects, and behavioral paradigm helps in developing and validating theories which may help in analyzing and estimating institutional or personal behavior. In IS research, theories mostly are applied to ‘artificial phenomena’ (March & Smith, 1995). These theories can be developed from two ways: i) they may be developed from evaluation and examination and ii) they may be created (March & Smith, 1995). The nature of design and behavioral research in information systems has been clearly explained by Hevner et al. (2004) through a conceptual framework (shown in

Figure 2-1 below), following which they have proposed a set of guidelines which may help the researchers to conduct, evaluate, and appreciate design research. The framework proposed by Hevner et al. (2004) is mainly divided into three blocks: environment, IS research, and knowledge base, which will be evaluated by relevance and rigor. Environment consists of people (which consists of roles, features and capabilities), organizations (which consists of culture, strategies and processes), positioned relative to technology (which in turn consists of architecture, capacity, applications, infrastructure and communications).

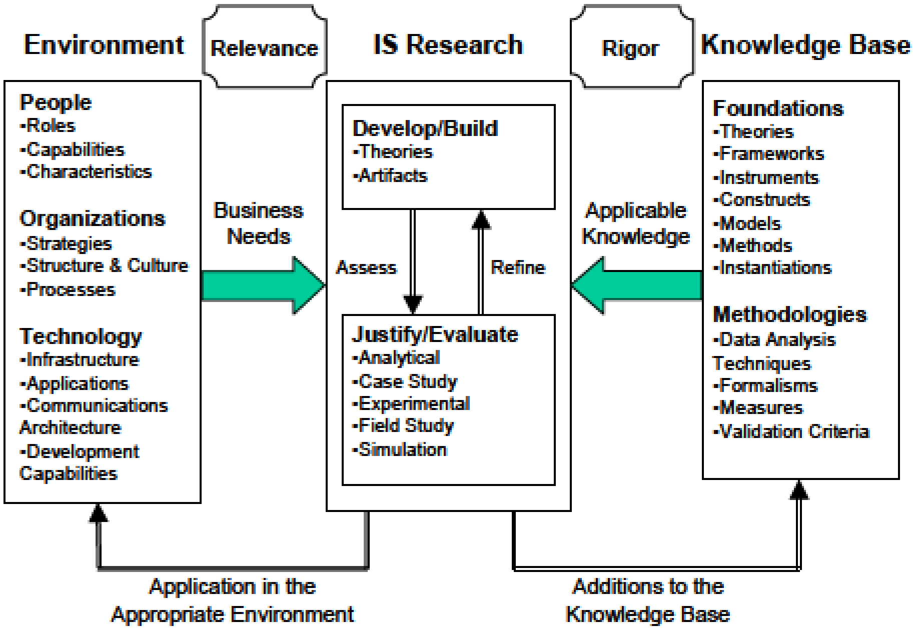


Figure 2-1 Information systems research framework (Hevner et al., 2004)

Whereas IS research involves develop/ build (theories and artifacts will be developed) and justify/evaluate (which will be done through experiments, field studies, simulations, and case studies); and knowledge base consists of foundations (which consist of theories, frameworks, instruments, constructs, models, methods etc.) and methodologies (which consist of measures, validation techniques, data analysis, techniques, and formalisms). Design science involves developing new artifacts based on the business needs, while behavioral science involves truth (Hevner et al., 2004). Theories will be evaluated based on how well they have addressed a problem and how useful are those contributions to the environment, and the design science and behavioral science researches will be evaluated based on their contribution to the body of knowledge base and how well they satisfy the business needs (Hevner et al., 2004). Hevner et al. (2004) indicates that intrinsically design science is a 'problem solving process' and main contribution of a design science research will depend on the artifact development, and therefore, they indicate that they have developed seven guidelines which can help a researcher in the process of addressing how to conduct a design science research. The seven guidelines proposed by Hevner et al. (2004) are:

i) Design as an artifact: Artifacts of design research and behavior science research are required for the inception of models, constructs, methods and instantiations.

ii) Problem relevance: Concentration of design research must be on a technology based solution which can help in procuring knowledge for those irresolute questions from past. Relevance of the solution to the problem is important.

iii) Design Evaluation: The design of an artifact must be properly authenticated based on their characteristics and application. Evaluation is very important for design science research. The requirements of business environment are evaluated based on the artifact. Evaluation provides the required comments on the design of the artifact to the construction phase, which will further help the construction phase to make any changes to the design process.

iv) Research contributions: the value of a design research is dependent on an appreciable contribution, i.e. "What are the new and interesting contributions?" A design science research must have all or at least one of these contributions: the design artifact, foundations, and/or methodologies.

v) Research Rigor: Research rigor defines the manner in which a research is regulated. It is defined independently in design science research and behavior science research. In design science research, rigor is required in the architecture and interpretation of the designed artifact, application of suitable metrics, etc. Coming to the behavior science research, rigor is estimated based on the relativity to the analysis methods and suitable data.

vi) Design as a Search Process: in the process of identifying a valid solution to a problem, design is necessary (Hevner et al., 2004). Search process as a part of design is crucial and also is a very difficult task. Finding an effective solution to a problem is what matters. Means (sets of actions), ends (goals and constraints), and laws (uncontrollable forces) are the three critical components of the design science research. All these are dependent on the environment and problem.

vii) Communications of Research: Research must not only be acquainted with technology but also must be acquainted with management oriented organizations. The process of artifact construction is to be known by the researchers who are involved with it, and also helps in forming research ideas for future.

2.3 Why Design Research Methodology

As new algorithm and a new artifact are developed, we can say that design science research methodology was used in this research work. Design science research focuses on developing artifact or model or method based on the business needs (Hevner et al., 2004). Design science helps in finding a way to accomplish human goals (March & Smith, 1995). In the process of paving a way, March & Smith (1995) have developed a two dimensional framework through which they indicate that design science is possible. The first dimension focuses on four research activities: build, evaluate, theorize, and justify; while the second dimension focuses on four design research outputs: constructs, models, methods, and instantiations (March & Smith, 1995). Design science uses mostly artificial phenomena’s (such as hypothesis) and artifacts, and the artifacts/methods/instantiations developed through design science will be considered as contributions to the body of knowledge (March & Smith, 1995). Peffers, Tuunanen, Rothenberger, & Chatterjee (2007) have presented the process to conduct design science research in five steps (seen in Figure 2-2): i) identifying the problem and motivation: which is the process of identifying research problem (proposal), ii) objectives of solution: which discusses the need for a better artifact, iii) development: which includes implementing the developed model or artifact, (iv) demonstration: finding a suitable context, v) evaluation: which involves evaluation of the model and feedback, and vi) communication.

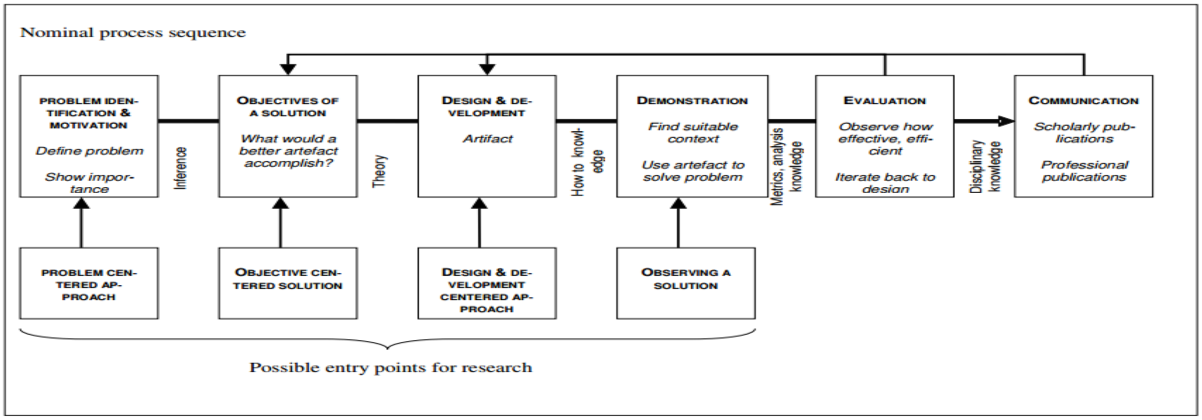

Nominal process sequence
PROBLEM IDENTIFICATION & MOTIVATION
Define problem
Show importance
Inference
OBJECTIVES OF A SOLUTION
What would a better artefact accomplish?
Theory
DESIGN & DEVELOPMENT
Artifact
How to knowledge
DEMONSTRATION
Find suitable context
Use artefact to solve problem
Metrics, analysis knowledge
EVALUATION
Observe how effective, efficient
Iterate back to design
Disciplinary knowledge
COMMUNICATION
Scholarly publications
Professional publications
PROBLEM CENTERED APPROACH
OBJECTIVE CENTERED SOLUTION
DESIGN & DEVELOPMENT CENTERED APPROACH
OBSERVING A SOLUTION
Possible entry points for research

Figure 2-2 Design research methodology process model (Peffers et al., 2007)

2.3.1 Identifying the problem

The process of identification and motivation a problem requires the researcher to demonstrate the need for that model, which will be motivated by problem centered initiation (Cole, Purao, Rossi, & Sein, 2005; Peffers et al., 2007). For this dissertation, the problem and motivation of the dissertation is discussed in the abstract.

2.3.2 Objectives of Solution

Objectives of solution includes the need to identify a better artifact, which is motivated by objective centered solution (Eekels & Roozenburg, 1991; Hevner et al., 2004; Peffers et al., 2007). This step can act as literature review. For this dissertation, the literature can be found in sections 1.1, 1.2, 1.3, 1.4, 3.2, 3.4, 4.2 and 4.3.

2.3.3 Development

Development involves the process of developing the artifact, which will be motivated by developing design and development centered initiation (Hevner et al., 2004; Peffers et al., 2007). For this dissertation, the developed artifacts can be found in sections 3.6 and 4.5.

2.3.4 Demonstration

Demonstration is the process in which the developed artifact will be implemented to solve the problem. This step requires an appropriate context or a client where the artifact can be implemented (Peffers et al., 2007). Demonstrating the artifact of this dissertation can be found in sections 3.7 and 4.6.

2.3.5 Evaluation

Evaluation step involves the process of analyzing the ability and productiveness of the artifact, based on which the researcher will think if he/she should proceed with their artifact or should revisit the design step (Peffers et al., 2007). Evaluation of the artifacts developed in this dissertation can be found in sections 3.8 and 5.1, 5.4.

2.3.6 Communication

Once the artifact is evaluated the researcher will enter communication step where, the knowledge acquired from this design research methodology can be shared with fellow researchers through scholarly publications and other journal resources (Peffers et al., 2007).

Chapter 3 ‘Analyzing workflows in business processes for obstructions due to authorization policies’ has been published in Hawaii International Conference for Social Sciences, 2012 (Spear, Malladi, & Lakkaraju, 2012). Parts of Chapter 4 and 5 have been published in Trustworthy Systems and their Applications (TSA) 2014 conference under the title ‘Integrated modeling and analysis of attribute based access control policies and workflows in healthcare’ (Lakkaraju & Xu, 2014). Mobile device adoption in healthcare has been published as a book chapter in the ‘Cases on healthcare information technology for patient care management’ (Lakkaraju & Lakkaraju, 2011). Context awareness has been discussed in MWAIS 2013 conference under the title ‘context aware knowledge management system in healthcare (Lakkaraju, 2013). The role of mobile technology in healthcare has been reviewed in the MWAIS 2011 conference under the title ‘Framework to investigate the role of mobile technology in healthcare organizations (Lakkaraju & Moran, 2011).

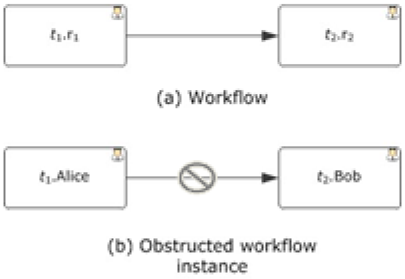
3 ANALYZING WORKFLOWS IN BUSINESS PROCESSES FOR OBSTRUCTIONS DUE TO AUTHORIZATION POLICIES

Chapter 2 is focused on the research methodology used in this dissertation: design science research methodology. In Chapter 3, the motivating scenario to develop an algorithm for analyzing obstruction free instances in workflows caused by authorization policies, existing literature on analyzing obstruction free workflow instances, various types of authorization constraints such as binding of duty and separation of duty, some definitions and symbols used in this chapter, the proposed algorithm, implementation evaluation and complexity analysis of the algorithm are going to be discussed.

3.1 Motivating Scenario

It is well-known that aligning security policies with business objectives is a difficult task. To address this, we present a new approach to analyze work-flow instances for obstructions due to static and dynamic authorization policies. We give a new algorithm that allows organizations to properly assign users to tasks without the policies causing obstructions (e.g. deadlocks). Our work is novel since we consider loops, conditions and parallelism in workflows, through a new concept called “release” events. We illustrate our approach on some real world workflows in healthcare and financial industries.

Workflows are used to describe the pattern of tasks to be executed by users to achieve business objectives. A workflow can be implemented in many ways, possibly unboundedly many ways, called instances. An instance has an obstruction if authorization policies make the instance invalid. For example, consider the workflow in Figure 3-1.(a). One of its instance in Figure 3-1.(b) has an obstruction, if either Alice or Bob is not allowed to play roles r1 and r2 respectively. It would also have an obstruction, if the authorization policy states that the tasks t1 and t2 must be executed by the same user.



(a) Workflow

(b) Obstructed workflow instance

Figure 3-1 An example of obstructions in workflows

It is not always possible to determine if a workflow instance has obstructions, with just casual inspection. Automated analysis is very much desirable, since it gives a high degree of confidence about the analysis. Literature concerning analysis of other similar infinite-state models such as cryptographic protocols is replete with security violations that could only be found with automatic and formal approaches (Lowe, 1996). This is the problem we consider in this paper. Our main contribution is an algorithm that answers the following question: "Given a workflow W, an instance ω of it, and a set of authorization policies E, is ω obstruction-free with respect to E?" We can use this in turn to answer the question, "If U is a set of users playing the roles R of tasks T in W , what are the set of substitutions of U to R that are obstruction-free?". This algorithm enables hospitals for instance, to implement workflows without obstructions, while securely enforcing the authorization policies. Though there has been considerable work reported in literature on this topic, our work is novel, since we allow workflows to have conditions, loops and parallelism. We justify the newness of our work more in Section 3.2 related work. We also have a small, fast Java implementation.

We will first discuss the related work and how this work is novel in section 3.2. In section 3.3, we will discuss various types of authorization constraints. In section 3.4, we will define some terms, followed by discussion on symbols used in section 3.5. In section 3.6, we will show the proposed algorithm. In section 3.7, with real time case studies we will discuss how the algorithm can be used to identify obstructed workflow instances with respect to user to role assignment. In section 3.8 we will evaluate the algorithm. In section 3.9, complexity analysis will be discussed, followed by discussion in section 3.10.

3.2 Related Work

Our algorithm decides satisfiability for bounded users in the sense of (Bertino, Ferrari, & Atluri, 1999; Tan, Crampton, & Gunter, 2004)'s paper, which states that a policy is satisfiable if there exists an assignment of users to tasks that does not violate the policy. It also decides if a workflow is "sound" in the sense of Van der Aalst (2004) who calls sound workflows as those that do not have dead transitions or deadlock before completing their final tasks. Most past works on checking authorization policies did not consider conditions, loops and parallelism in workflows including (Bertino et al., 1999; Ligatti, Bauer, & Walker, 2005; Sandhu, 1988; Schneider, 2003; Tan et al., 2004; Van der aalst, 1998). The only other work that considers conditions, loops and parallelism is the recent work of Basin, Burri, & Karjoth (2011). Some points to note in comparison are: (i) They give interesting theoretical results based on their framework on CSP (Roscoe, 2005), while our work is applied: We have a practical implementation of an algorithm based on their concepts, and we have applied it to some commonly used workflows in the real-world; (ii) Our algorithm is based purely on substitutions and nodes in the workflow. We do not claim that our approach is better or worse than (Basin et al., 2011). However, like many constraint-satisfaction approaches, it is easier to develop a practical implementation; (iii) They demonstrate that finding out whether a workflow can have an obstruction-free instance with respect to a set of policies is decidable but NP-Hard, by reducing the problem to graph-coloring. Ideally, one should run their algorithm to first to check if a set of policies are enforceable, and then use our algorithm to find if a particular instance is obstruction-free; (iv) The concept of "release points" used in our algorithm was first introduced by Basin et al. (2011), extending previous work on security-annotated graphical workflow models (Wolter, Schaad, & Meinel, 2008).

3.3 Authorization Constraints

An authorization constraint is put into a system in order to control the access to particular components of the system. This access is based upon an authorization policy that defines the actions that are allowed in the given system. There are three types of authorization constraints that we consider here, in regards to workflows: static authorizations, dynamic Separation of Duties, and dynamic Binding of Duties. Dynamic constraints remain in effect until the workflow reaches a release point, or release event, a concept first introduced by Basin et al. in (Basin et al., 2011). When the workflow encounters a release event, all previous associations between users and tasks in the associated security policy are removed. In our workflows, a release event is represented by a person exiting a door, adapted from Basin et al. (2011).

Static authorizations describe a security policy that simply maps each user with a set of permissible tasks. This policy prevents users with inappropriate clearance from executing tasks above their authorization level, demonstrated in Figure 3-6, showing a visual representation of the user-role-task authorization mappings. We can see that each user is associated with an authorization level, which is then in turn tied to specific tasks which are allowed to be executed by users at that level. For example, Mary is a doctor, and as such is allowed to perform only task 1 (identifying diagnostic requirements). Mary does not, however, have access to the patient's insurance records, and cannot then perform tasks 3 or 4. This must be done by a user authorized to execute these tasks, such as Jones or Dennis. In this paper, static authorizations are not outlined in the workflow models, yet will be defined before-hand as a set of user-task assignments, given as a relation U T = {(t.u) | t ∈ T , u ∈ U } where U represents the set of users, T represents the set of tasks. In this case, if (t.u) ∈ U T, then we say that the user u has static authorization to perform task t.

Dynamic Separation of Duties (SoD) defines a widely accepted security policy enforced to prevent fraud in the system by ensuring that no single user can access all components in the workflow (Ferraiolo, Sandhu, Gavrila, Kuhn, & Chandramouli, 2001). This system works by preventing conflicts of interest within the workflow. For example, consider the scenario of a purchase order being placed within a business, shown in Figure 3-4. The task of placing the order, t1, is in conflict with the task of approving the order, t2, since a user wanting to commit fraud could both place and approve a phony order, while simply pocketing the money. SoD constraints would prevent any user previously associated with executing t1 from executing t2.

We use the notation s = (T1 , T2 , oj) to represent an SoD constraint, where T1 and T2 are disjoint sets of tasks, and oj represents a particular release event tied to s. In our workflow models, SoD constraints are represented by the "≠" notation. This symbol is associated (via a dotted line) to the task or grouping of tasks that are constrained by a SoD constraint. In the above healthcare workflow example, Figure 3-5 contains the authorization constraint s1 = (t1 , t2 , o1), meaning that any user who executes t1 cannot then execute t2 until release event o1 is reached, removing all associations between users and tasks involved in s1 that have been made up to that point. It is important to note that the placement of release points is crucial to the meaning of the workflow. For example, in Figure 3-5, if release point o2 had instead been placed on the ‘yes’ path, the user that performed t3 would then be released from executing t4, which would not force the two tasks to be executed by the same user.

Dynamic Binding of Duties (BoD) is a policy put into place in order to control the number of users executing particular tasks. For example, in a workflow in which sensitive data is required to perform a task t1, and the same sensitive data is required to execute a subsequent (though not necessarily successive) task t2 , the authorization policy will bind the tasks in such a way that they must be executed by the same user. In a similar notation to that used for SoD constraints, we represent a BoD constraint as a tuple b = (T , oj) such that any user who executes a task in T is then exclusively bound to all tasks in T, and no other user is allowed to execute these tasks until release point oj has been reached. For the workflows in this paper, we represent BoD constraints with an “=” symbol, linked to a set of tasks, T, by a dotted line.

For example, in Figure 3-5, tasks t3 and t4 are bound with a BoD constraint, indicating for instance that if Dennis was to execute t3, he must then be the one to execute t4, and no one else. If, however, Dennis was to execute t3, and patient is not insured, such that the next event is the release event o2, then any user statically authorized to execute t3 can do so without constraint. As explained in the motivating scenario, authorization constraints on a workflow can interfere with the implementation of the workflow by causing potential deadlocks. The workflow given in the Figure 3-1 is a trivial one. Hence, it is easy to detect an obstruction in an instance of it, simply by manual inspection. However, workflows such as the healthcare workflow (Figure 3-5) in Section 3.7 are complicated. Real-world workflows are even more complicated. It is difficult to precisely determine if a given assignment of users to roles results in obstructions in such workflows, since there are many tasks, users, conditions, parallelism and loops. Automated analysis of such workflows is very much necessary to gain assurance that their instances are obstruction-free under a given set of user to role assignments.

3.4 Definitions

In this section we will define some terms that are used in this study.

Definition 1: A workflow template is a sequence of events, starting with a start, ending with a finish, and any number of (t.r) events in between where ‘t’ belongs to the set of tasks and r belongs to the set of roles.

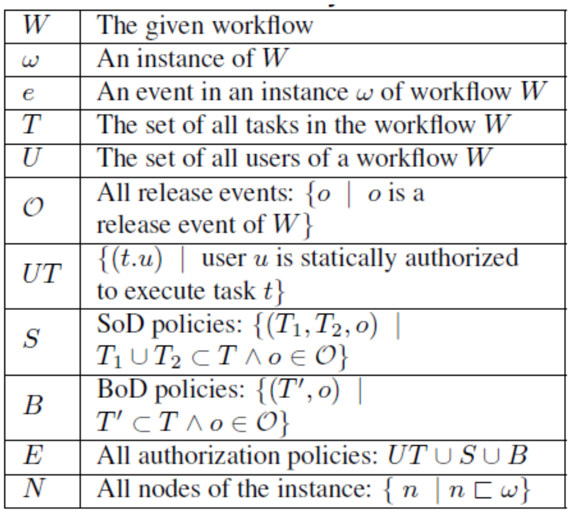
Definition 2: A workflow instance is an instantiation of a workflow template by assigning users to roles. Formally, ω is an instance of a workflow template W, if there is a substitution σ of users to tasks that can be applied on W to yield ω. i.e., $W \sigma = \omega$.

From now on, we will just call a workflow template simply a workflow.

3.5 Symbols

The symbols we use in our algorithm are given in Table 3-1.

Table 3-1 Symbols



W	The given workflow
ω	An instance of W
e	An event in an instance ω of workflow W
T	The set of all tasks in the workflow W
U	The set of all users of a workflow W
$\mathcal{O}$	All release events: $\{o \mid o$ is a release event of $W\}$
UT	$\{(t.u) \mid$ user u is statically authorized to execute task $t\}$
S	SoD policies: $\{(T_1, T_2, o) \mid T_1 \cup T_2 \subset T \land o \in \mathcal{O}\}$
B	BoD policies: $\{(T', o) \mid T' \subset T \land o \in \mathcal{O}\}$
E	All authorization policies: $UT \cup S \cup B$
N	All nodes of the instance: $\{ n \mid n \sqsubset \omega\}$

Some points to note

- The release event o_j corresponds to a unique security policy in E. The notation t_n and u_n will be used to refer to the task and user of the node n respectively. This does not mean that the task number must correspond to the user number, but simply that t_n refers to the task in the node n, and u_n refers to the user in node n;

- This algorithm uses an array L of size |T| where T is the set of all tasks. The purpose of this array is to track which users have performed which tasks in order to check that no security obstructions occur. The array will be set up in such a way that the n*th* position in the array will correspond to the n*th* task in the workflow. Thus, for example, when a $t_a.u_b$ event is encountered in ω, if no security obstruction is found, the a*th* position in L will be set to u_b;

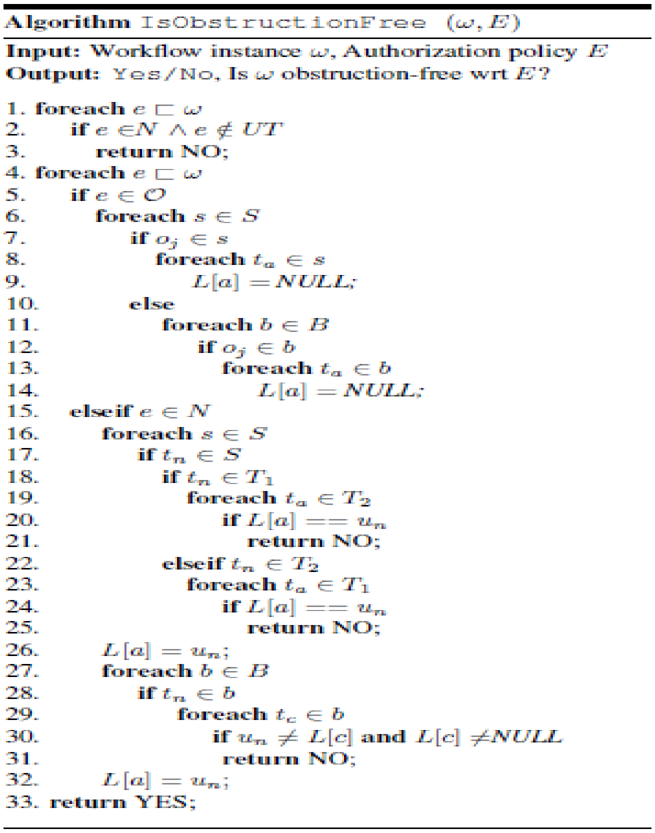
- We write e ‹ ω if e is an element of the sequence ω.

- The assumption is made that the given instance ω is a valid instance of the workflow, ensuring the existence of a start and finish event, no skipped or extra tasks, and also proper flow (ensuring the workflow is followed in a proper direction).

• The assumption is also made that the policies given are valid polices. For instance, a policy that both binds t1 and t2 and also separates the two is not considered. In this case, however, our algorithm would simply return 'NO' at the first encountered obstruction.

3.6 Algorithm

The proposed algorithm is shown below.



```
Algorithm IsObstructionFree $(\omega, E)$
Input: Workflow instance $\omega$, Authorization policy $E$
Output: Yes/No, Is $\omega$ obstruction-free wrt $E$?

foreach $e \sqsubset \omega$
    if $e \in N \wedge e \notin UT$
        return NO;
foreach $e \sqsubset \omega$
    if $e \in O$
        foreach $s \in S$
            if $o_j \in s$
                foreach $t_a \in s$
                    $L[a] = NULL;$
            else
                foreach $b \in B$
                    if $o_j \in b$
                        foreach $t_a \in b$
                            $L[a] = NULL;$
    elseif $e \in N$
        foreach $s \in S$
            if $t_n \in S$
                if $t_n \in T_1$
                    foreach $t_a \in T_2$
                        if $L[a] == u_n$
                            return NO;
                elseif $t_n \in T_2$
                    foreach $t_a \in T_1$
                        if $L[a] == u_n$
                            return NO;
        $L[a] = u_n;$
        foreach $b \in B$
            if $t_n \in b$
                foreach $t_c \in b$
                    if $u_n \neq L[c]$ and $L[c] \neq NULL$
                        return NO;
        $L[a] = u_n;$
return YES;
```

Figure 3-2 Algorithm for user to role assignments for obstruction free workflows

We will use the collateral evaluation workflow given as a running example in Basin et al. (2011) (Figure 3-3). Using this workflow helps in justifying that our approach can handle workflows with release events that were handled by another approach in Basin et al. (2011). The workflow is used by a financial institution to approve the acquisition of the collateral provided by the borrower.

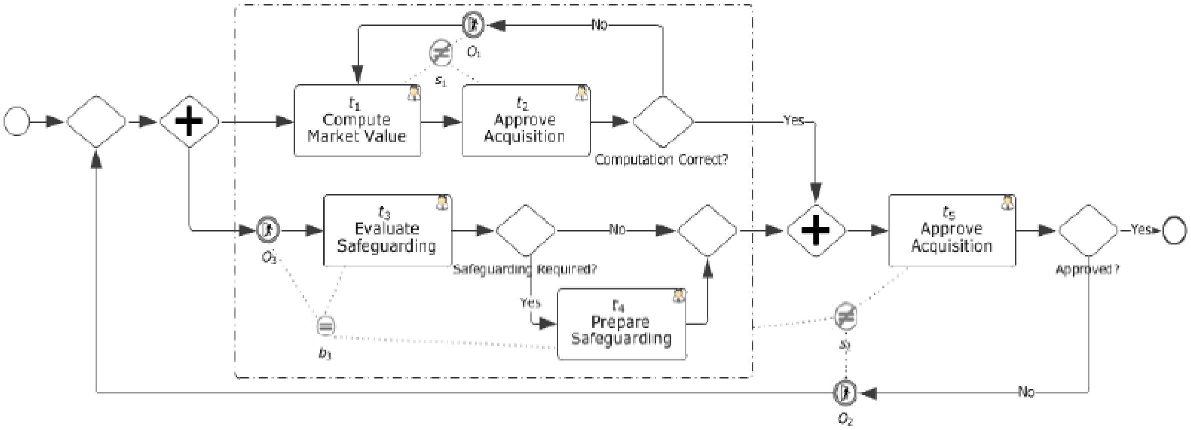


Figure 3-3 Financial Industry: Collateral evaluation workflow from Basin et al.

Note that in this workflow:

(a) Tasks t1 and t2 belong to SoD policy s1;

(b) Also, tasks t3 and t4 are grouped into the BoD policy b1 , meaning that these tasks must be performed by the same user;

(c) Lastly, all four of these tasks are grouped into the SoD policy s2. The static authorization policy is also given in [2].

The complete set of constraints is as follows:

$$
\begin{aligned}
s_1 &= \{ \overbrace{\{t_1\}}^{T_1}, \overbrace{\{t_2\}}^{T_2}, o_1 \} \\
s_2 &= \{ \overbrace{\{t_1, t_2, t_3, t_4\}}^{T_1}, \overbrace{\{t_5\}}^{T_2}, o_2 \} \\
b_1 &= \{\{t_3, t_4\}, o_3\} \\
UT &= \{(t_1.Alice), (t_2.Alice), (t_5.Alice), (t_2.Bob), \\
&\quad (t_3.Bob), (t_4.Bob), (t_1.Claire), (t_2.Claire), \\
&\quad (t_1.Dave), (t_2.Dave), (t_3.Dave), (t_4.Dave), \\
&\quad (t_5.Dave)\}
\end{aligned}
$$

Consider the following instances of the workflow:

ω1 = (t1 .Alice, o3, t3 .Bob, t2 .Bob, o1, t1 .Alice, t4 .Bob, t2 .Claire, t5 .Dave) ω2 = (t1 .Alice, o3, t3 .Bob, t2 .Alice, o1, t1 .Bob, t2 .Claire, t4 .Bob, t5 .Claire)

We first trace through our algorithm with the obstruction-free instance ω1. The first step of the algorithm is to check that each node in the instance is in U T. For ω1, this is easily verifiable. After this check is completed, assuming that no obstructions have occurred, the first event reached is the node n = t1 .Alice. There are 9 steps that the algorithm goes through in order to process this event.

1. Check if the event is a release event.

2. Since it is not, we check that this event is a node.

3. It is, so we enumerate through each SoD policy si to check if the task of the node is involved in si.

4. We find that t1 is involved in s1 and s2.

5. For s1, we check which set of tasks that t1 is a part of. This way, we know which tasks are meant to be separated from t1.

6. We check that the corresponding entries in L for each task in T2 are not set to the user of the node, Alice.

7. Since L [2] = Alice, there is no obstruction for this event relative to security policy s1 , and we set L[1] = Alice.

8. We repeat steps 5 - 7 for s2, this time comparing
'Alice' to L [5] = NULL.

9. Check if t1 is involved in any BoD policy. Since it is not, we move on to the next event in ω.

The next event encountered is the node n = t3 .Bob, which follows the same pattern of steps above to result in setting L [3] = Bob. The difference in the above step with this event is with step 9. We find that t3 is indeed involved in a BoD policy, b1. The algorithm then follows these steps to process this:

9. Check if t3 is involved in a BoD policy.

10. It is, so we enumerate through each BoD policy bi to check if the task of the node is involved in bi.

11. We find that t3 is involved in b1.

12. We check that the corresponding element in L to each of the tasks in b1 is either the same user of the node we are looking at, Bob in this case, or NULL, meaning no other user has executed these tasks that are bound together.

13. Since L [3] = Bob and L [4] = NULL, there is no obstruction for this event relative to the security policy b1.

14. Since t3 is not involved in any other BoD policies, we have no need to repeat steps 11 - 13, and move on to the next event in ω.

The next event in ω1, t2 .Bob, is a similar operation to previous events. At this point, the non-NULL elements in L are L [1] = Alice, L [2] = Bob, and L [3] = Bob. We then encounter the release event o1, which is tied to tasks t1 and t2, so these entries in the array L are set to NULL. The three subsequent events result in L [1] = Alice, L [2] = Claire, L [3] = Bob, L [4] = Bob. The last event, t5 .Dave is involved in s2, and in the set T2, so we check each element in L corresponding to t1, t2, t3, and t4 to be sure that Dave has not executed any of these tasks in T1. Since none of the four elements in L is equal to 'Dave', we set L [5] = Dave and return 'YES', indicating that this instance is obstruction-free.

For ω2, the first three events are the same from ω1, so using the steps above, after the third event in ω2, the non- NULL elements in L are L [1] = Alice and L [3] = Bob. The next event in ω2 is t2 .Alice. We see that t2 is involved in the security policy s1 and s2. We first check for violations in s2. Since $t2 \in T2 \subset s1$, we check the corresponding element in L of each task in T1 for a conflicting assignment of tasks to users. In this case, L [1] = Alice, which makes the event t2 .Alice a violation, since t1 and t2 are not authorized to be executed by the same user. Thus, this instance is not obstruction free, and the algorithm returns 'NO'.

3.7 Implementation

3.7.1 Business purchase approval scenario

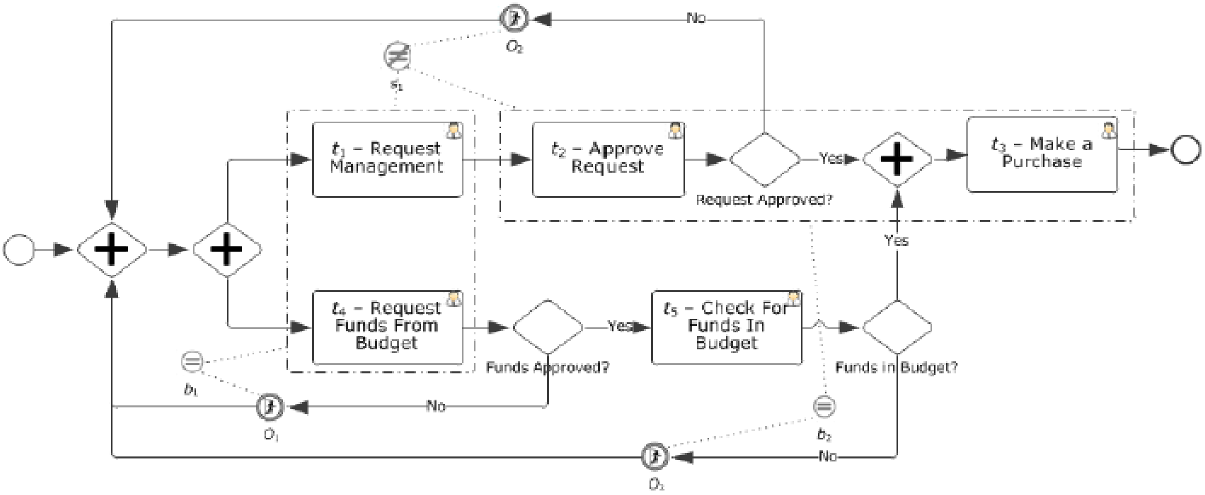


Figure 3-4 Purchase approval workflow

Figure 3-4 is an example workflow for a business purchase. In this example, Task 1 is to first send the request to the proper management and at the same time to execute Task 4, being to request the funds from the budget. Task 2 is to have the request from Task 1 approved and Task 5 verifies that the funds are in the budget. When all of these tasks have been completed, Task 3 is to actually make the purchase. Note that Tasks 2 and 3 are grouped, and connected by BoD policy b1 and release event o1 , indicating that Tasks 1 and 4 must be done by the same user, and cannot be performed by another user until release event o1 is reached. The next group of tasks, 2 and 3, are linked to this first group of tasks (t1 and t4) by SoD policy s1 and release event o2. This policy ensures that the user that is bound to executing tasks 1 and 4 is then prohibited from executing tasks 2 and 3. This second group is also a part of the BoD policy b2 with release event o3. The complete set of constraints is as follows:

$$\begin{aligned} s_1 &=\{ \overbrace{\{t_1,t_4\}}^{T_1}, \overbrace{\{t_2,t_3\}}^{T_2}, o_2 \} \\ b_1 &= \{\{t_1,t_4\},o_1\} \\ b_2 &= \{\{t_2,t_3\},o_3\} \end{aligned}$$

$$UT = \left\{ \begin{array}{c} (t_1.Alice),\ (t_4.Alice),\ (t_1.Bob), \\ (t_2.Bob),\ (t_3.Bob),\ (t_4.Bob), \\ (t_1.Claire),\ (t_4.Claire),\ (t_5.Claire), \\ (t_1.Dave),\ (t_2.Dave),\ (t_3.Dave), \\ (t_4.Dave),\ (t_5.Dave) \end{array} \right\}$$

Our first example implementation of our algorithm is using this workflow. Consider the following instances of the workflow:

ω1 = (t1 .Alice, t2 .Bob, o2, t1 .Alice, t2 .Claire, t3 .Claire, t4 .Alice, t5 .Alice).

ω2 = (t1 .Bob, t2 .Bob, t3 .Bob, t4 .Dave, t5 .Claire)

The first instance contains three static authorization obstructions, which would be found in line 2 of the algorithm since the events t5 .Alice, t2.Claire, and t3.Claire are nodes that are not a members of the set U T. If static authorization obstructions do exist in a given instance, these obstructions are caught early in the algorithm, since these user-task assignments are not considered valid, and therefore render the remaining events in the instance irrelevant.

ω2 contains an SoD obstruction, which the algorithm finds in line 24 since t2 ∈ T2 and for t1 ∈ T1 , L[1] = Bob = un , meaning that Bob had already executed a task in a conflicting set of tasks. Though the algorithm will have returned ‘NO’ already, this instance also contains a

BoD obstruction with the event t4.Dave, since t1 was executed by Bob, and these two tasks are meant to be bound.

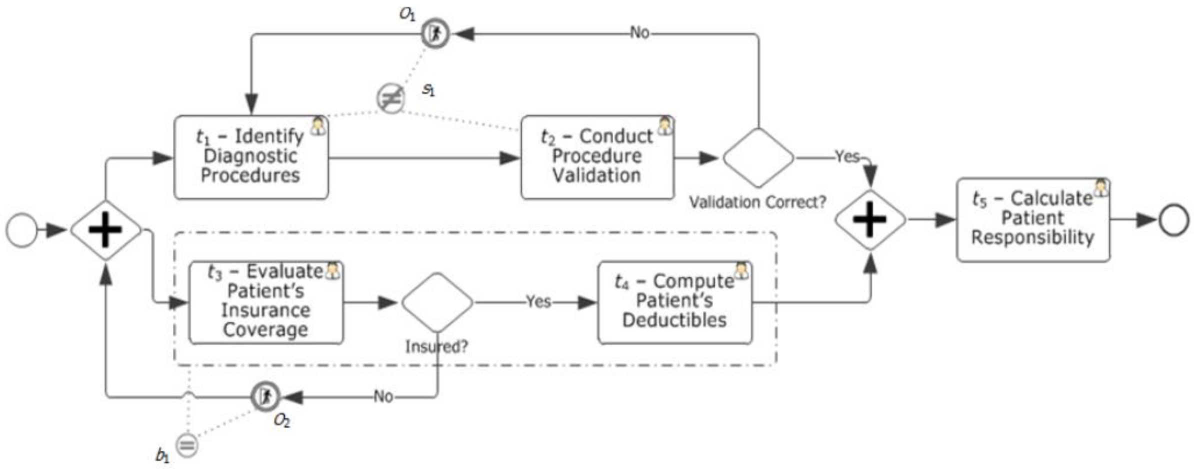
3.7.2 Healthcare Scenario

Figure 3-5 is an example of healthcare which depicts the patient's deductible workflow. The role to task assignment is shown in the following Figure 3-6. The following 3 sample instances can be retrieved:

Instance 1: ω1 = (t3 .Dennis, o2, t1 .Mary, t2 .Dennis, t3 .Jones, t4 .Dennis, t5 .Jones).

Instance 2: ω2 = (t1 .Tresa, t2 .Dennis, o1, t3 .Dennis, t1 .Mary, t2 .Dennis, t4 .Dennis, t5 .Jones).

Instance 3: ω3 = (t1.Tresa, t2.Dennis, t3.Jones, o2, o1 , t1.Mary, t2 .Dennis, t3.Jones, t4.Jones, t5.Jones).



o_1
No
s_1
t_1 - Identify Diagnostic Procedures
t_2 - Conduct Procedure Validation
Validation Correct?
Yes
t_5 - Calculate Patient Responsibility
t_3 - Evaluate Patient's Insurance Coverage
Insured?
Yes
t_4 - Compute Patient's Deductibles
No
o_2
b_1

Figure 3-5 Patients deductible workflow

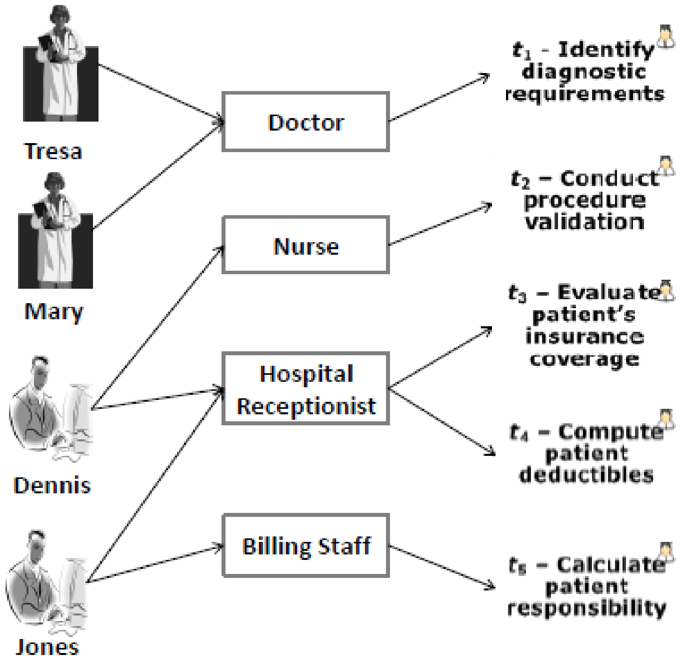

Tresa
Mary
Dennis
Jones
Doctor
Nurse
Hospital Receptionist
Billing Staff
t_1 - Identify diagnostic requirements
t_2 – Conduct procedure validation
t_3 – Evaluate patient's insurance coverage
t_4 – Compute patient deductibles
t_5 – Calculate patient responsibility

Figure 3-6 Role-Task assignment

ω1 contains a BoD obstruction with the event t4 .Dennis. The algorithm will take several steps to catch this obstruction, first noting that, in the previous event, since no obstruction will occur with the execution of t3 by Jones, the corresponding entry in L will be set to L [3] = Jones. However, when Dennis attempts to execute t4, our algorithm will check the BoD policies that t4 is included in to be sure that the sets of tasks in each policy are being executed by the same user. In this case, if a user other than Dennis is listed in L as having per- formed t3, then this is an obstruction. We see in L that Jones executed t3, and therefore the BoD constraints are violated in this instance ω1. Thus, the algorithm returns 'NO', and the obstruction is correctly identified.

In the second instance, ω2, there are no obstructions. This is properly identified by our algorithm after lines 1-3 find that each node in ω2 is in U T , indicating that no static authorization obstructions occur, after lines 4-14 'reset' the values in L for the tasks corresponding to the event o1 , after lines 16 - 26 verify that each event in ω2 involved in an

SoD policy is not in conflict with any previous events, tracked in the array L, indicating no SoD obstructions, and finally after lines 27 - 32 verify that each node in ω2 involved in a BoD policy is not in conflict with any previous tasks.

We use the last example instance ω3 as an example of the ability of our algorithm to handle the loops and conditions of a workflow. This instance follows all release events in the workflow, and tests each possible value of each condition.

3.8 Evaluation

The workflow ‘financial industry: collateral evaluation workflow from Basin et al., (2011)’ (Figure 3-3) has been used as a running example in section 3.6. The workflow ‘purchase approval workflow’ (figure 3-4) is used to show separation of duties (under section 3.3), and the same has been used in implementation (section 3.7). And the last healthcare workflow ‘patients deductible workflow’ (figure 3-5) has been used to pick instances; and those instances are used for evaluation. The evaluation of instance 1 and instance 3 has been shown below. Instance 1 has a binding of duties obstruction, whereas instance 3 involves all of the loops properly handling all of the release events and outputs that the instance 3 is free of obstructions.

Output for Instance 1: ω1 = (t3 .Dennis, o2, t1 .Mary, t2 .Dennis, t3 .Jones, t4 .Dennis, t5 .Jones) can be seen below:

```
C:\Program Files\Java\
jdk1.6.0 26\bin>java IsObstructionFree
Checking for static authorization
violations ...
```

```
        No static obstructions found.

        Checking for dynamic obstructions ...

        For event t3.Dennis, checking for policies involving t3. Found t3 in s1. Checking
for obstructions. No SoD obstruction found, setting L[1] = Dennis.

        For event o2, Releasing all events.

        For event t1.Mary, checking for policies involving t1. Found t1 in s1. Checking
for obstructions. No SoD obstruction found, setting L[2] = Mary.
```

After displaying similar statements for other events, the trace then ends with:

```
        For event t3.Dennis, checking for policies involving t3. Binding of Duties
obstruction found for this instance! Obstructions found for this instance!

        C:\Program Files\Java\jdk1.6.0 26\bin>
```

For instance 3: ω3 = (t1.Tresa, t2.Dennis, t3.Jones, o2, o1 , t1.Mary, t2 .Dennis, t3.Jones, t4.Jones, t5.Jones), the algorithm correctly returns that no obstructions are found, properly handling the operations of each release event. The actual output of the program evaluating instance 3 is given below:

```
C:\Program Files\Java\
jdk1.6.0 26\bin>java IsObstructionFree                    -
Checking for static authorization
violations ...
No static obstructions found.
Checking for dynamic obstructions ...
```

```
For event t1.Tresa, checking for policies involving t1. Found t1 in s1. Checking for
obstructions. No SoD obstruction found, setting L[1] = Tresa.
For event t2.Dennis, checking for policies involving t2. Found t2 in s1. Checking for
obstructions. No SoD obstruction found, setting L[2] = Dennis.
```

After displaying similar statements for other events, the trace then ends with:

```
For event t5.Jones, checking for policies involving t5. No dynamic obstructions found.
No obstruction found for this instance!
C:\Program Files\Java\jdk1.6.0 26\bin>
```

When we change ω3 slightly such that in the last three tasks, t3 and t4 are now performed by Dennis, the algorithm still finds the instance to be obstruction-free, again showing the ability to handle loops and conditions. If these release events were not handled properly, the event t3.Jones would have caused one of these changes in events to violate BoD policy b1, however, because of our algorithm's novel ability to handle these events; this instance is correctly found to be obstruction-free.

3.9 Complexity Analysis

The input to the algorithm is rather simple: In addition to the workflow and the policies, it is just a set of user/role substitutions. This allows even naive users of our implementation to easily analyze their workflow instances. The implementation itself is just three pages of Java code and it has successfully output all the results presented in this paper. The best-case scenario for the algorithm is a workflow with no release events. i.e., there are no dynamic policies (SoD, BoD) in the authorization policies, only static, if any. Instance ω2 of the purchase-approval workflow is such a case. Obviously, in this case, the complexity is O(n) where n is the number of events, since only the first loop between lines 1 to 3 is executed and the second loop on line 4 is executed as well, but without any statements inside it. The worst-case scenario is when every release event that is tied to a BoD policy is encountered at least once. Instance ω3 of the healthcare workflow is one such case. In this case, lines 4 through 14 are executed for all the events, resulting in a complexity of O (n4).

The average-case is when there are release events in the workflow that are not encountered in the instance. In this case, the complexity is O (n3). Instance ω1 of the Purchase Approval workflow is one such.

3.10 Discussions

An immediate extension of this work is to determine all the set of substitutions of users to roles for a given set of users. To accomplish this, we can enumerate all possible substitutions of users to roles and use our main algorithm for each substitution. The complexity of this would be obviously proportional to the number of users. An algorithm given by Basin et al. (2011) to solve the same problem has polynomial complexity when the number of users is large and the static authorizations are well-distributed. It is possible that our approach would have similar complexity under similar conditions. A detailed complexity analysis is our task on hand.

We would like to improve our algorithm by adding the ability to check for inconsistent policies, such as a policy stating that a task should not be executed by the same user, while another states that it should be executed by the same user. We would also like to allow any number of tasks in our implementation (the current version allows a maximum of ten).

4 INTEGRATED ANALYSIS OF WORKFLOW AND ACCESS CONTROL

4.1 Accommodating BOD and SOD in the Methodology

Chapter 3 is focused on identifying obstruction free workflow instances of a given workflow due to BOD and SOD constraints. Automated analysis of workflow instances to identify any obstructions due to access control policies is very much desired, which is not always possible through causal inspection. Once all workflow instances are considered to be free of obstructions, it can be considered that no access control policy is obstructing the workflow. It can also be a case where access control policies are inefficient in covering all the workflow activities. Chapter 3 has discussed about determining if a workflow instance is obstruction-free, however it is not possible to identify which policies are covering which workflow activities and it is also not possible to identify if all of the workflow activities in a workflow instance are being covered by the access control policies. This chapter focuses on addressing these specific issues by developing an integrated methodology to identify workflow activities that are not being protected by access control policies and thereby improving the existing access control policies. Given a workflow, each workflow activity is identified in terms of subject, resource, action, and environment attributes. Subject attributes include role, user id, age, phone number, address, etc. Based on a subject’s role, we will identify the resources that the subject can access; after that, user to resource allocation can be identified. Following that, binding of duty constraints and separation of duty constraints can be validated as discussed in chapter 3 for any obstructions in the workflow instances. Once the workflows are identified to be obstruction free, integration of workflow and access control is done using the proposed methodology, which in turn will be used to identify if there are any workflow activities that are not being covered by access control policies, using which we can also identify if there are any unused policies, and thereby can improve the access control policies and workflow activities.

4.2 ABAC

Workflows can be very complicated, especially in a complex environment like healthcare which may involve various subjects trying to perform actions on certain resources in multiple environments, thus requires controlling the access of resources by subjects. For a subject to perform an action on a resource in an environment, that subject should be authorized to perform the intended action. In this research, Business Process Modeling Notation (BPMN) has been used to develop workflows, which is discussed in chapter 1.1. Inter-organizational systems allow users to access and share data beyond organizational boundaries and therefore needs proper authorization mechanisms to protect sensitive information from being exposed to unauthorized personnel. An access control policy defines the conditions to which access to resources can be granted and to whom (Ferreira et al., 2007). With the increasing complexity of information systems, access control methods have evolved from Mandatory Access Control (MAC), Discretionary Access Control MAC (DAC), Access Control List (ACL), Role-Based Access Control (RBAC), to Attribute-Based Access Control (ABAC) (Xu & Zhang, 2014). ABAC is “an access control method where subject requests to perform operations on objects are granted or denied based on assigned attributes of the subject, assigned attributes of the object, environmental conditions, and a set of policies that are specified in terms of those attributes and conditions” (Hu et al., 2013). ABAC is developed from RBAC. The preferability of ABAC over other access control mechanisms is demonstrated in detail in (Yuan & Tong, 2005). In ABAC, the authorization elements are defined in terms of attributes, rather than identities, of subjects, actions, resources, and environments (SARE). Attributes are characteristics of these entities that are predefined and pre-assigned by an authority. ABAC basically involves subject attributes, action attributes, resource attributes, and environment attributes. Subject attributes help in identifying a user (or an operation or a process) with its characteristics rather than its role; for example: job title, user name, age, etc. A resource attribute is a characteristic of a resource (record, data, etc.); for example: medical elements of patient record, non-medical elements of patient record, etc. Action attributes help in identifying the type of action the subject will perform on a resource; for example: read, write, etc. Environment attributes are operational or situational characteristics, such as current time and IP address. A sample organizational ABAC implementation has been shown in Figure 4-1. It mainly involves certain functional 'points': Policy Enforcement Point (PEP), Policy Decision Point (PDP), Policy Administration Point (PDP), and Policy Information Point (PIP). Considering a particular scenario where a subject requests access to a resource under certain environment, this is sent as a request into a policy enforcement point (PEP) where a decision is taken whether to 'allow' that access or 'deny' the access based on certain attributes (user, action, resource, and environment). PEP sends that request to PDP where the request is broken into subject attributes, resource attributes, action attributes and environmental attributes.

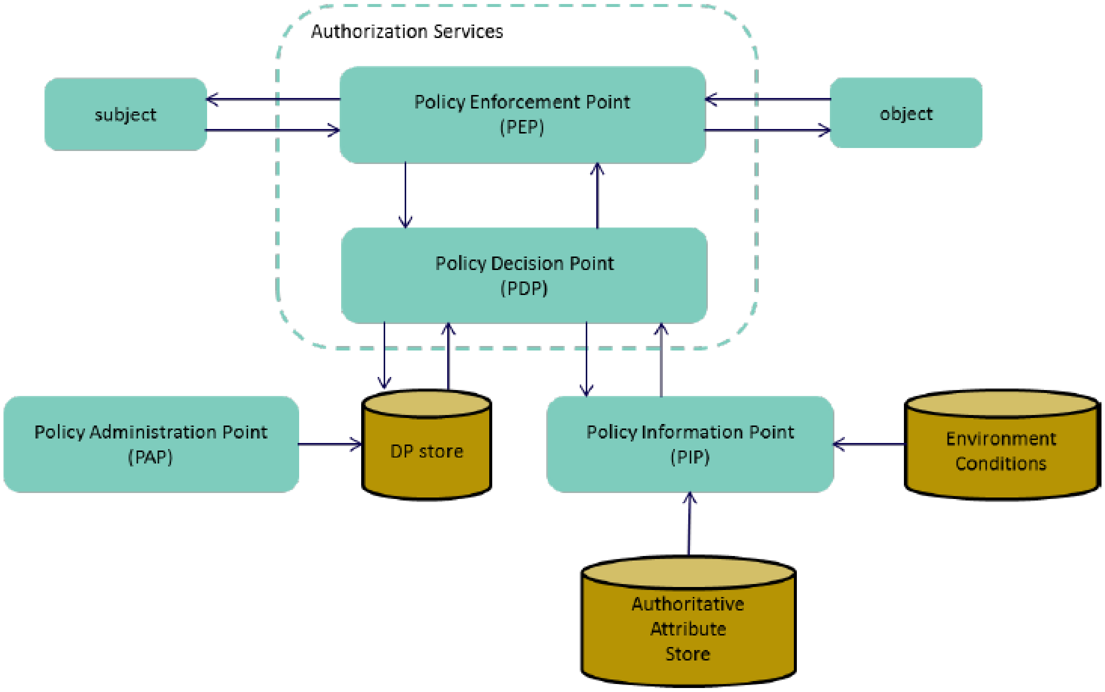


Figure 4-1 Access control mechanism distribution in a sample organizational ABAC implementation (Hu et al., 2013)

Initially, each of those attributes is stored in their respective attribute stores managed at the policy information point (PIP). And the Policy Administration Point (PAP) is the point where policies are managed by the administrators. PDP will validate the request based on the available policies from PAP and the attributes available from PIP. Once the PDP is able to validate the request, it will send a notification to PEP suggesting to 'allow' the access, otherwise 'denies' the request. To support ABAC, OASIS has developed the eXtensible Access Control Markup Language (XACML) standard (Identifier, 2005). Consider the following access control rule: *Any doctor can read and write into patient health record (PHR) only for his/her designated patients from internal network*. The subject attributes include 'doctor', the action attribute is 'read' and 'write', and the resource attribute is 'patient health record', and the environment variable is 'internal network' and the condition is that the doctor can only access the health records of the patients designated to him/her. In XACML, the above rule can be written as shown in Appendix [1]

4.3 Literature Review

A workflow involves execution of a series of certain activities which help in achieving a goal (Chaari et al., 2004). These workflows must be executed in a secure way, which is possible through access control. To achieve these secured workflows, various access control techniques have been studied in accordance with the workflows, but none of the works have shown an integrated analysis and have only dealt with workflows and access controls separately. In (Chaari et al., 2004), authors have extended role based access control and proposed an authorization and access control model for workflows. The access control model proposed by Chaari et al. (2004) allows certain users who possess certain roles to access a resource only when the subjects are executing certain tasks. Research by Chaari et al. (2004) mainly focuses on the user's role and the task the user has to execute; but it does not consider any other contextual information or user attributes except the role. Russello et al. (2008) have proposed a workflow based access control framework for e-health applications, where permission is granted to a user based on the necessity of the task that the users have to execute. Russello et al. (2008)'s research mainly focuses on the 'need' of the user to execute the task, but does not consider any of the environment or resource or subject attributes except the role. Russello et al. (2008) indicates that 'access rights must be granted only to entities that require access to a given resource and just for the amount of time that access is necessary', but healthcare is rather dynamic environment where access is required in varying contexts which requires precise contextual information to allow a user to access certain resource. This dissertation will address this issue by focusing on importance of 'environment' attributes and our methodology will demonstrate the integrated analysis of workflow and access control which can grant access rights only to the entities that require access to a given resource under certain context where access is necessary. Le, Doll, Barbosu, Luque, & Wang (2012) have improved the traditional role based access control model to enhance the information access management in the environment of workflow and team collaboration in healthcare programs. Le et al.'s research has mainly concentrated on traditional RBAC and is an extension of RBAC. This dissertation deals with ABAC, which not only deals with user attributes such as 'role' but also with various resource, environment and action attributes. Lu, Zhang, & Sun (2009) have developed a framework to specify security constraints in workflow by improving business process management notation model. Authors have improved the workflow model to aid role based access, separation of duty, history based task assignment, etc. Compared to this work, our model deals with the attributes of the subjects, action, resource, and environments rather than role based or SOD or history based or task based. Zhang & Liu (2011) have developed a workflow oriented attribute based access control model that provides 'Service-Oriented Computing', where any device's functionality can be offered as a standard service. Although Zhang & Liu (2011) talk about attribute based access control, they have not considered the need to identify unused policies or the need to identify unprotected workflow activities. XiangPeng, Cerone, & Krishnan (2006) have developed a framework integrating workflows (BPEL) and access control (RBAC) to execute security constraints through 'temporal logic'. Though the framework deals with integrating workflows and RBAC through temporal logic; regular people may not exercise the logic to execute security restraints. Compared to their work, our model deals with enforcing ABAC policies on the workflow activities which is not covered in (XiangPeng et al., 2006).

None of the above mentioned literature talks about integrated analysis of workflow and ABAC policies. One of the main takeaways from our research is that it is possible to identify any unused policies and also it is possible to identify any workflow activities that are not being protected by access control policies.

4.4 Assumptions

Some of the assumptions in developing the current model are discussed below. We are not developing a new modeling language. We are adding a new element 'environment' to the activity of the workflow. We are suggesting breaking down an activity in the workflow in terms of 'Subject, Action, Resource, and Environment' (SARE) elements as shown in Figure 4-2. Not every activity may have all of the four (SARE) elements, i.e. each activity may have any of the SARE elements. We identify common themes between the workflow and ABAC policies in terms of SARE elements. In any given situation to integrate a workflow and policies, we assume that workflow and policies will have any or all of the SARE elements. We assume workflow is a valid one without any deadlocks. Organization will define the SARE elements to be used commonly by both workflow and the ABAC policies.

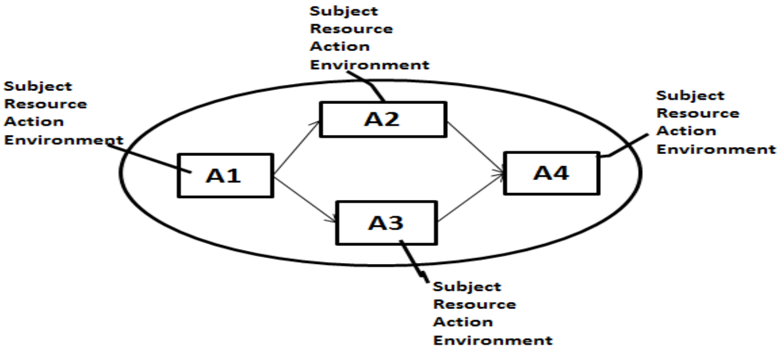


Figure 4-2 Workflow activity breakdown

4.5 Methodology

The main contribution of integrated modeling and analysis of workflows and access control policies is that it helps in identifying the workflow activities that are not being protected by access control policies and also helps in recognizing unused access control policies thereby helps the administrators in improving the existing access control policies or developing new policies. The general idea of integrated modeling and analysis is as follows: (a) specify workflow activities in terms of SARE elements, (b) specify ABAC policies, (c) cross-examine the workflow activities and ABAC policies to identify discrepancy, (d) if there is no discrepancy, then terminate, otherwise go back to (a) if the discrepancy is about the workflow activities, or go back to (b) if the discrepancy is about the ABAC policies. In the following, we focus on the cross-examination of workflow activities and ABAC policies, as illustrated in Figure 4-3.

Step 1: The given workflow may include many activities (A1, A2, A3…). Each activity has to be individually identified from the given workflow as (A1), (A2), and (A3) ….

Step 2: From the individual activity that has been identified from the workflow (say A1), subject resource action and environment (SARE) attributes have to be identified, i.e. each activity (A1) has to be broken down into SARE (subject action resource and environment) elements (Figure 4-3's break the activity in terms of SARE).

Step 3: ABAC policies will be written targeting the Subject and/or Resource elements of the SARE attributes. An ABAC policy is represented by the target which the policy is intended to cover. It is seen in Figure 4-3 where P1 is represented by R1, where R1 is the resource1 targeting any workflow activity which involves the resource1. All the workflow activities that involves resource1 (R1) will be covered by this policy. Therefore, in this step, the target element from the policy is extracted and used to represent the policy to match with the respective attribute in the workflow activity.

Step 4: Once the workflow activities are broken into SARE elements, and the targets are extracted from the Policies, each of the 'subject and resource' attributes of the workflow activity are cross examined across the available ABAC policies that are targeting the respective subject and/or resource attributes.

Step 5: It is important to identify all the policies that can be applied on an activity. Once the policies are applied on the activity of the workflow, record each of those policies per workflow activity into a table (Table 4-1). Identify all of the policies that are being applied on each of the activity of the workflow and develop a table (Policies applicable on each activity of the workflow: Table 4-1).

Step 6: Once the policies are applied on the activity of the workflow, identify all of the policies that are not being used on the workflow (Policies that are not being used on the workflow: Table 4-2).

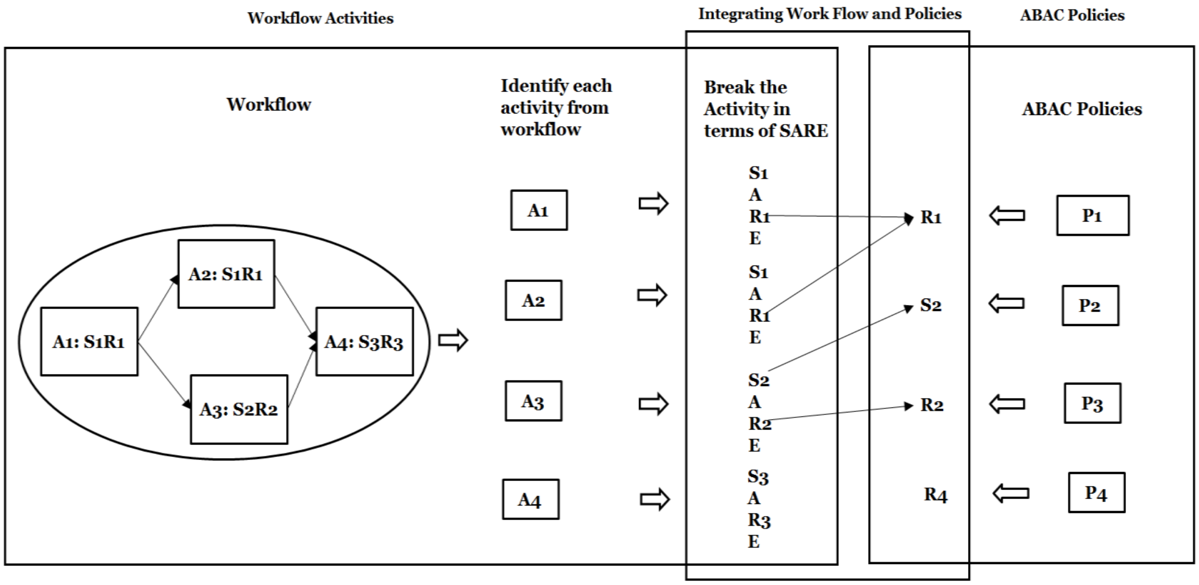


Figure 4-3 Model to integrate workflow and ABAC policies

4.6 Implementing the approach

Figure 4-3 represents the integrated model of the workflow and ABAC policies. It is evident from the Figure 4-3 that the ‘workflow’ denotes a workflow with four activities: A1, A2, A3, and A4. Each of these activities has ‘subject, resource, action, and environment attributes’. Once there is the workflow, each of the activities of the workflow is extracted from the workflow as shown under the ‘Identify each activity from workflow’ task in Figure 4-3 ({A1}, {A2}, {A3}, and {A4}). Following this, break each of these activities into the subject, action, resource, and environment elements (SARE).

{A1} => {S1, A, R1, E}; {A2} => {S1, A, R1, E}; {A3} => {S2, A, R2, E}; and {A4} => {S3, A, R3, E}

Similarly, from the given ABAC policies {P1, P2, P3, and P4}, identify individual policies. Each of these policies will further include multiple rules which are written targeting either a subject or a Resource. Based on the target, each of these policies may be targeted to either subject or resource. Therefore, each of these policies can be shown as under ‘ABAC Policies’ in Figure 4-3, P1=>R1, P2=>S2, P3=>R2, P4=>R4. Once the subject and resource elements are found from the workflow activities, the policies with respective subject or resource targets must be imposed on that particular activity as shown below.

As {A1} has subject S1 and Resource R1, any of the policies that target Subject S1 or Resource R1 can be imposed. From the given policies, only P1 is applicable. This case is a special example where there is no policy specified on subject S1, and therefore this activity of the workflow is **incompletely covered (partial)** by the access control policy. It is the same case with A2.

Whereas {A3} has a subject S2 and Resource R2, and therefore, policies P2 and P3 can be applied on the activity A2 showing that A2 is completely covered.

Coming to the activity {A4}, it has subject S3 and Resource R3, but there is no policy targeting either subject S3 or Resource R3. Therefore, this activity is not at all covered. Similarly, Policy P4 is targeting resource R4, which has no occurrence in the workflow, resulting in no use of this policy. The outputs from the current model are shown in the below Tables 4-1 and 4-2. Based on these outputs, the workflow developers and access control policy administrators can make the necessary changes to the respective workflows and policies.

4.7 Expected results of the integration model

Table 4-1 Policies applicable on each activity of the workflow

Activity Name	Policy applicable (Yes/ N0/ Partial)?	If applicable, Policy Name
A1	Partial	P1
A2	Partial	P1
A3	Yes	P2, P3
A4	No	-

Table 4-2 Policies in use (Y/N)

Policy	Policy in use (Y/N)?
P1	Y
P2	Y
P3	Y
P4	N

Based on these outputs from Tables 4-1 and 4-2, it is evident that (i) Activities A1 and A2 are each covered by one Policy P1, whereas Activity A3 is covered by two policies P2 and

P3, (ii) The workflow administrators that the ‘IOS workflow activity A4 is not being covered by any ABAC policy and therefore it is recommended to re-evaluate the activity A4, (iii) the Policy P4 is not covering any workflow activity and therefore it needs re-evaluation, (iv) in re-evaluation, policy administrators may re-evaluate the uncovered workflow activity and the policies that are not in use, and based on the need, they may suggest the organizations to add new SARE elements using which the administrators may develop new policies or workflow activities.

5 CASE STUDY ON INTEGRATED METHODOLOGY

Chapter 4 is focused on the methodology to integrate workflow and access control policies. Chapter 4 also discussed some background and importance of attribute based access control (ABAC) mechanism, sample organizational implementation of ABAC, why ABAC is preferred for this dissertation compared to other access control mechanisms, some assumptions in developing the methodology, the proposed model to integrate workflow and access control policies, demonstration of implementing the approach, and expected results of the integrated model. Following which, chapter 5 focuses on a case study in healthcare scenario where the proposed methodology is implemented and studied.

5.1 Demonstrating the Methodology in a Healthcare Scenario

Manipal hospital is located in Bangalore, India with 54 bed count and can accommodate up to 91 beds when fully equipped. It is located in the center of the city. The hospital has six departments: administration department, inpatient/outpatient department, pharmacy department, laboratory department, emergency department, and surgical ward. The hospital accommodates both inpatients and outpatients. The pharmacy department provides medications for both the inpatients and outpatients. The scientific staff of the hospital include: front desk specialists, doctors, nurse specialists, lab specialists, pharmacists, and surgical specialists. This hospital has recently implemented and mandated to use electronic health records across the hospital. The main parts of the electronic health records may include: administrative record, clinical record, pharmacist record, patient record, and surgical record. Administrative record of the EHR mainly includes patient demographic information (such as age, sex, first and last names, email addresses, physical addresses, etc.), date of admission and discharge, patient registration information, etc. This administrative record is restricted to be accessed only by the front desk personnel, nurse specialists and doctors. The clinical record stores the health related information of the patient. The clinical record may include detailed information of pathology reports, blood reports, ENT reports, doctor notes, nurse notes, prescriptions, etc. The pharmacist record includes prescription ids, patient ids, physician ids, and pharmacist notes. The surgical record of the EHR includes the surgical information, images (like Xrays, etc.), surgery specialist, and patient information. The subject, action, resource, and environment (SARE) attributes are provided by the hospital and are discussed below.

Subject attributes may include: department id (which includes an administration department, pharmacy department, emergency department, etc.); specialty id (which includes doctors, nurses, pharmacists, etc.); user id (which includes patient ids, doctor ids, nurse ids, etc.). Action attributes may include: read, write, update, delete, fill, validate, approve, reject, etc.; resource attributes include: electronic health record, patient health record, pharmacist record, clinical record, administrative record, surgical record, etc.; environment attributes include: ip addresses (92.92.1.1, 93.93.1.1, etc.), inpatient ward, outpatient ward, laboratory, surgery ward, doctor office, etc. Each of the departments of the hospital is assigned with different IP address as mentioned below: for hospital administration department: 92.92.1.1; for inpatient and outpatient wards, doctors' offices, and nurse quarters, the IP range is from 91.91.1.1 to 91.91.1.5; the internet protocol address for pharmacy department is 93.93.1.1, and for laboratory department is 94.94.1.1, for emergency department is 95.95.1.1, and for surgical ward is 91.91.1.6.

This case study contains multiple workflows which are developed for various causes. A basic scenario in a hospital starts when a patient walks in for an appointment with the doctor (Figure 5-2). This workflow mainly involves five workflow activities. Subjects involved in this workflow are patient and front desk person (FDP). FDP uses a system which has been configured for the IP address 92.92.1.1 and the FDP is allowed only to use this system from this IP address. The FDP cannot access any other systems configured with other IP addresses. Also the FDP can only access administrative record but not any other clinical information of the patient’s electronic health record. When a patient walks in, the front desk person (FDP) asks the patient for his/her demographic information and tries to retrieve the patient information, if the patient is coming in for the first time, the FDP will request the patient to fill in the registration form and FDP validates the form, and thereby registers the patient into the hospital system followed by scheduling an appointment with a doctor.

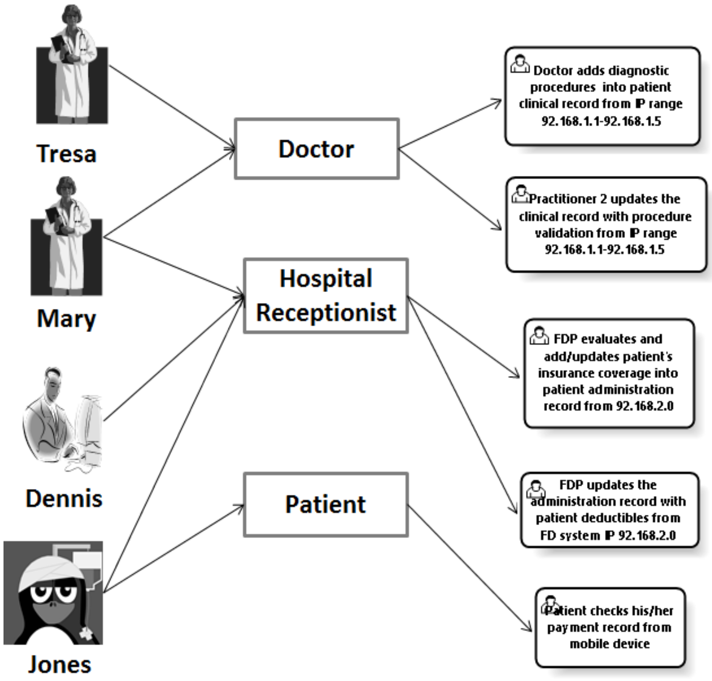

Tresa
Mary
Dennis
Jones
Doctor
Hospital Receptionist
Patient
Doctor adds diagnostic procedures into patient clinical record from IP range 92.168.1.1-92.168.1.5
Practitioner 2 updates the clinical record with procedure validation from IP range 92.168.1.1-92.168.1.5
FDP evaluates and add/updates patient's insurance coverage into patient administration record from 92.168.2.0
FDP updates the administration record with patient deductibles from FD system IP 92.168.2.0
Patient checks his/her payment record from mobile device

Figure 5-1 Subject attributes to resource allocation

This particular scenario can be considered as an example for separation of duties (SOD) and also binding of duties (BOD). There are mainly two roles: patient and FDP. A patient will check for an appointment with the FDP. FDP will check if the patient is valid, and then checks for doctor's schedule, and schedules an appointment. For this example, separation of duties applies between patient and receptionist. Main assumptions for a given scenario of patient's appointment with doctor to be static SOD are: (i) receptionist cannot be the patient; (ii) patient himself/herself cannot act as receptionist. If a receptionist is a patient, the receptionist cannot perform tasks checking the validity of the patient, checking for doctor's schedule and scheduling an appointment. Only another receptionist can perform those tasks.

As shown in the Figure 5-1, subject attributes to activity allocation can be seen, where multiple users with usernames (such as Tresa, Mary, etc) are allocated to subject attributes (role) and are thereby connected to their respective workflow activities. Following this, various workflow instances can be created and validated using the algorithm from chapter 3 using which the 'obstructed instances' be identified in the workflows. Once there is any occurrence of obstructions, the workflow activities should be modified and the workflow instances should be tested again to make sure that there is no occurrence of obstructed workflow instances.

As shown in the Figure 5-1, it is evident that activities 'FDP requests for patient demographic information...' and 'Patient fills the patient registration form...' cannot be completed by the same subject, accommodating separation of duty constraint. Similarly, activity 'FDP validates patient info in the admin record…' and activity 'FDP schedules appointment time with doctor…' should be done by the same subject, accommodating binding of duties constraint.

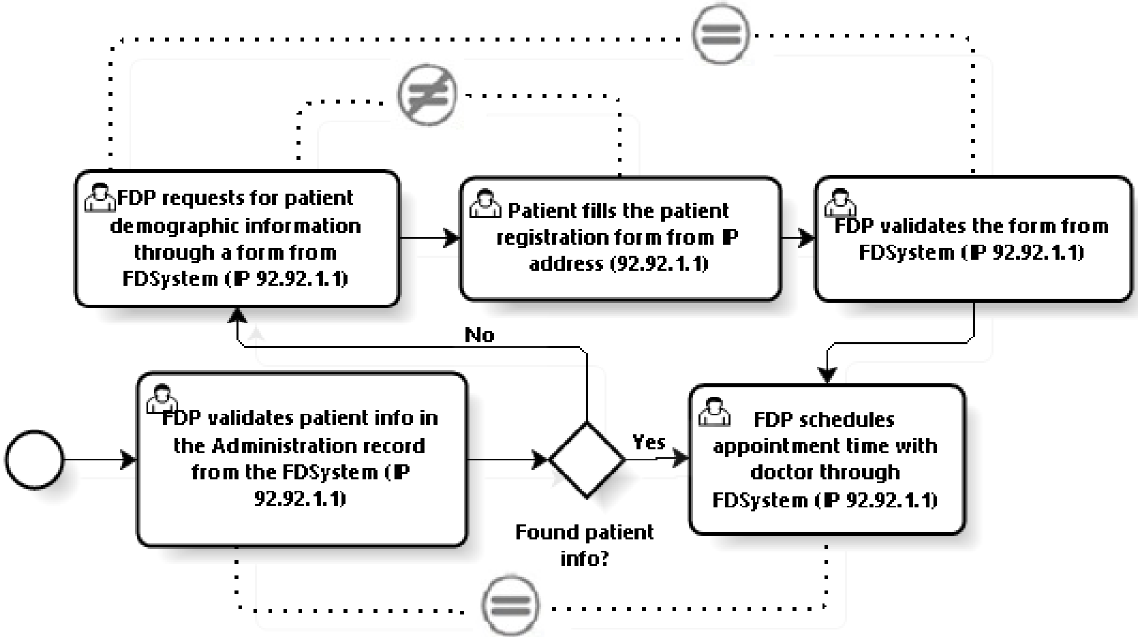

FDP requests for patient demographic information through a form from FDSystem (IP 92.92.1.1)
Patient fills the patient registration form from IP address (92.92.1.1)
FDP validates the form from FDSystem (IP 92.92.1.1)
No
FDP validates patient info in the Administration record from the FDSystem (IP 92.92.1.1)
Yes
FDP schedules appointment time with doctor through FDSystem (IP 92.92.1.1)
Found patient info?

Figure 5-2 Patient walks in for an appointment

Once the patient has an appointment with the doctor, the workflow in Figure 5-2 will begin. Activities of the workflow given in Figure 5-2 are discussed in detail. This particular scenario begins when a patient has an appointment with a practitioner and the practitioner recommends a diagnostic procedure.

This workflow starts with (activity A1) identifying the requirements for the diagnostic procedures to be performed (e.g. tests to evaluate a patient's health). Here the doctor will identify the tests and treatments for the patient from ip address 91.91.1.1-91.91.1.5;

Once the diagnostic procedures are identified, the recommended diagnostic procedures will be validated and performed in accordance with the Medicaid/Medicare regulations. This activity (A2) involves conducting the procedure validation to check for appropriate diagnosis in comparison with the Medicaid/Medicare regulations by another practitioner2 from 91.91.1.1-91.91.1.5. After that, if the procedure validation is correct and performed, the workflow will be carried on to Activity 4, otherwise diagnostic procedures will be reviewed and will be recommended to re-assess by the practitioner1 based on the Medicaid/Medicare regulations;

Then the patient's insurance will be evaluated by the front desk person of the hospital staff (e.g. by a receptionist) (A3) from ip address 92.92.1.1. Based on the patient's information, his/her insurance will be evaluated to verify whether his/her current insurance policy would cover that particular test (for example, an eye or a dental exam). If the insurance covers that particular test, workflow will move on to the next activity (A4). If the insurance will not cover the test, the patient may go back and check for a secondary insurance policy which will cover that test;

Activity 4 (A4) is where the financial administrator will calculate the patient's deductibles, once the patient's insurance is validated. This includes the percentage of the total bill that the insurance company would bear (e.g. 85%-15% deductible) vs the amount the patient is liable to pay.

Activity 5 (A5) is where the patient will access his/her payment record. The computed quote is available in the payment record and the patient will access it through a mobile device.

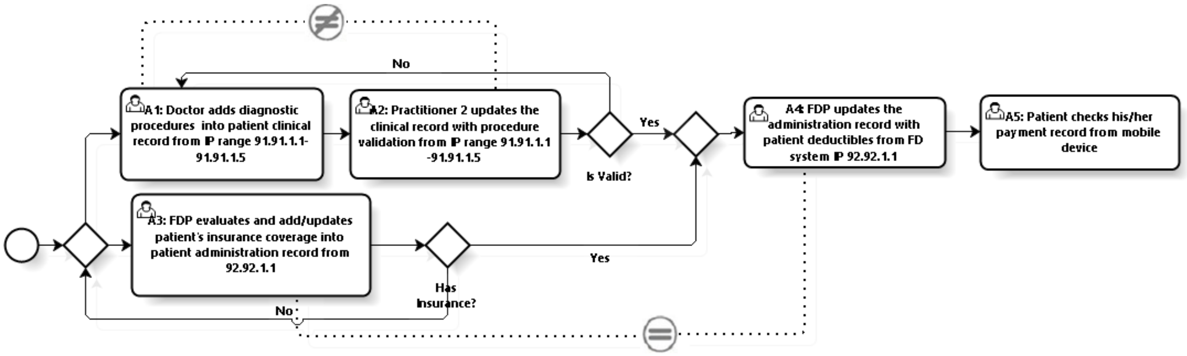

No
A1: Doctor adds diagnostic procedures into patient clinical record from IP range 91.91.1.1-91.91.1.5
A2: Practitioner 2 updates the clinical record with procedure validation from IP range 91.91.1.1 -91.91.1.5
Yes
Is Valid?
A4: FDP updates the administration record with patient deductibles from FD system IP 92.92.1.1
A5: Patient checks his/her payment record from mobile device
A3: FDP evaluates and add/updates patient's insurance coverage into patient administration record from 92.92.1.1
Yes
Has Insurance?
No

Figure 5-3 Patient diagnosis, insurance validation and bill payment

Insurance validation discussed in this particular scenario (Figure 5-3) involves both separation of duties and dynamic binding of duties constraint. The activities 'doctor adds diagnostic procedures and doctor 2 performs procedure validation' cannot be performed by the same doctor. Coming to the binding of duty, restricted subjects must have common actions, and subject should assume both activities at the same time. Patient will provide his/her health insurance to the receptionist from his/her mobile device. Receptionist will send the health insurance information on to health insurance inspector (or another receptionist) to validate the health insurance with valid health insurance company from FDP ip address 92.92.1.1. For this particular case, same subject with attribute: receptionist should validate the insurance and update the patient deductibles into the payment record from ip address 92.92.1.1.

Various other healthcare scenarios have been discussed in the figures below. Figure 5-3 is when a patient gets a notification about his/her appointment with the doctor, patient visits the hospital. In the hospital, FDP asks the patient to fill out a consent form. At the same time FDP asks the patient for his/her health insurance information and updates the health insurance information (if there are any changes). Patient will fill the form through a mobile device. These two tasks are bound by separation of duty constraint. Consent form cannot be filled by the FDP for the patient. Once the consent form is filled by the patient, the nurse will conduct initial assessment (weight, BP, etc.) and adds the assessment form and nurse notes to the clinical record. Once the initial assessment is done, nurse will send the nurse record to the doctor for pre-appointment analysis, where the doctor will retrieve the patient record and conduct pre-appointment analysis from 91.91.1.1-91.91.1.5.

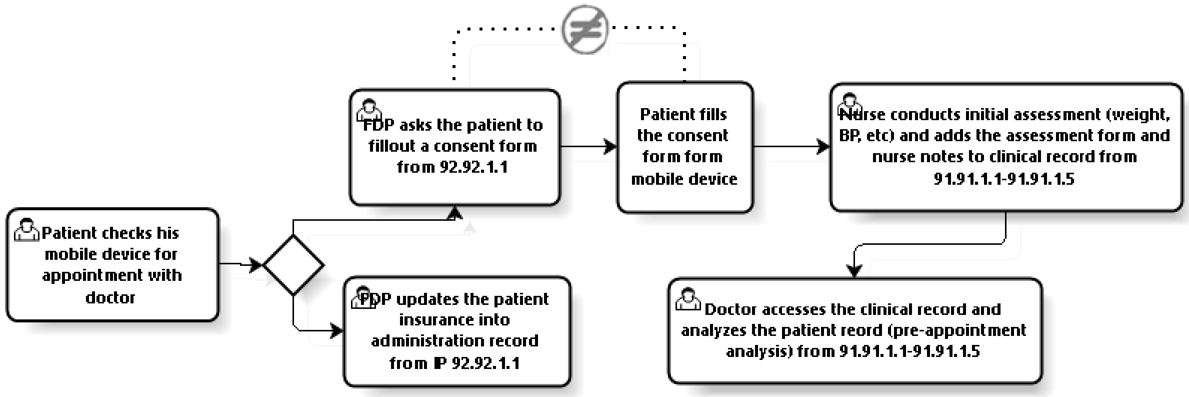

FDP asks the patient to fillout a consent form from 92.92.1.1
Patient fills the consent form form mobile device
Nurse conducts initial assessment (weight, BP, etc) and adds the assessment form and nurse notes to clinical record from 91.91.1.1-91.91.1.5
Patient checks his mobile device for appointment with doctor
FDP updates the patient insurance into administration record from IP 92.92.1.1
Doctor accesses the clinical record and analyzes the patient reord (pre-appointment analysis) from 91.91.1.1-91.91.1.5

Figure 5-4 Patient walk-in and nurse analysis

At the time of initial assessment, nurse may order any lab tests if he/she determines are necessary. The lab test orders are sent to the lab supervisor who will assign a lab specialist to perform lab tests, and the test reports will be added to the clinical record of the patient EHR and is forwarded to the doctor. Once the pre-appointment analysis is done, the doctor will review the lab reports and may recommend surgery for the patient, or may order any other lab tests, or may just give some health related suggestions. This entire scenario can be seen in Figure 5-4.

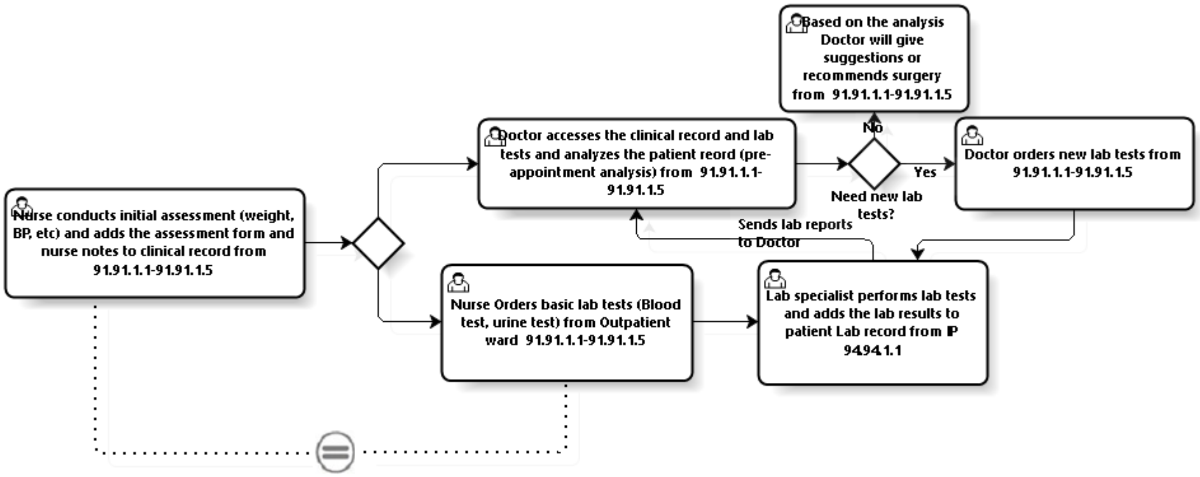

Based on the analysis Doctor will give suggestions or recommends surgery from 91.91.1.1-91.91.1.5
Doctor accesses the clinical record and lab tests and analyzes the patient reord (pre-appointment analysis) from 91.91.1.1-91.91.1.5
No
Yes
Need new lab tests?
Doctor orders new lab tests from 91.91.1.1-91.91.1.5
Nurse conducts initial assessment (weight, BP, etc) and adds the assessment form and nurse notes to clinical record from 91.91.1.1-91.91.1.5
Sends lab reports to Doctor
Nurse Orders basic lab tests (Blood test, urine test) from Outpatient ward 91.91.1.1-91.91.1.5
Lab specialist performs lab tests and adds the lab results to patient Lab record from IP 94.94.1.1

Figure 5-5 Generating lab reports

Based on the analysis, doctor may recommend the patient to get admitted into the hospital for further review, or doctor may recommend for an outpatient surgery. If there is no need for surgery, doctor may recommend some health related suggestions. If there is a recommendation of outpatient surgery, the patient may want to get a second review from another doctor, and separation of duty constraint in this case can be seen in Figure 5-5, where doctor 1 and doctor 2 cannot be the same person. But if patient decides to go with the surgery, the doctor may perform the surgery and update the patient record from IP address 91.91.1.1-91.91.1.5.

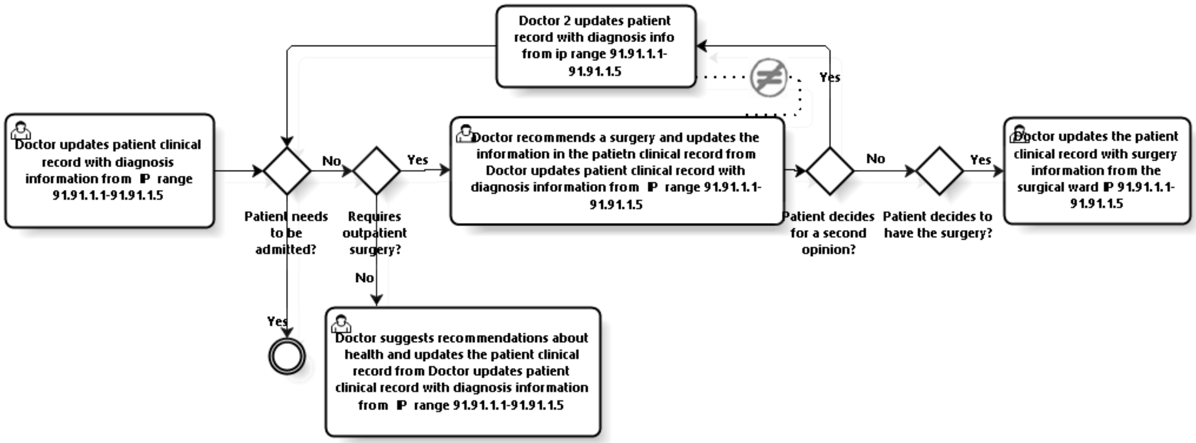

Doctor 2 updates patient record with diagnosis info from ip range 91.91.1.1-91.91.1.5
Yes
Doctor updates patient clinical record with diagnosis information from IP range 91.91.1.1-91.91.1.5
No
Yes
Doctor recommends a surgery and updates the information in the patietn clinical record from Doctor updates patient clinical record with diagnosis information from IP range 91.91.1.1-91.91.1.5
No
Yes
Doctor updates the patient clinical record with surgery information from the surgical ward IP 91.91.1.1-91.91.1.5
Patient needs to be admitted?
Requires outpatient surgery?
Patient decides for a second opinion?
Patient decides to have the surgery?
No
Yes
Doctor suggests recommendations about health and updates the patient clinical record from Doctor updates patient clinical record with diagnosis information from IP range 91.91.1.1-91.91.1.5

Figure 5-6 Doctor's recommendation for a surgery

Once the patient is admitted as inpatient or is being diagnosed as outpatient, nurse will enter the patient behavioral information into the nurse notes. The supervising doctor will review the nurse notes, and if all the information is accurate, the doctor will sign the nurse notes, otherwise may suggest some changes.

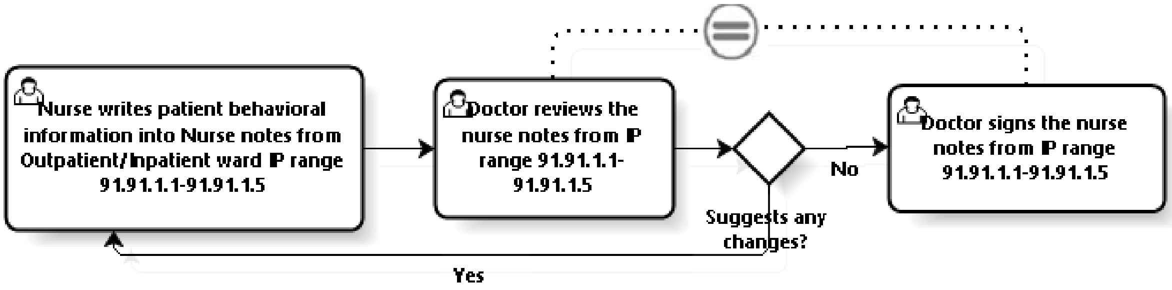

Nurse writes patient behavioral information into Nurse notes from Outpatient/Inpatient ward IP range 91.91.1.1-91.91.1.5
Doctor reviews the nurse notes from IP range 91.91.1.1-91.91.1.5
No
Doctor signs the nurse notes from IP range 91.91.1.1-91.91.1.5
Suggests any changes?
Yes

Figure 5-7 Nurse notes review

Nurse notes review in this particular scenario (Figure 5-7) can be considered as an example for dynamic binding of duties (BOD). The doctor who reviews the nurse notes should only sign the nurse notes, and two different subjects cannot be authorized to do these tasks independently.

Based on the patient's condition nurse may write prescription, which would be reviewed by the supervising doctor, and if doctor approves, the prescription will be forwarded to the pharmacist. Otherwise, doctor may edit the prescription and then ask the nurse to forward the prescription to the pharmacy.

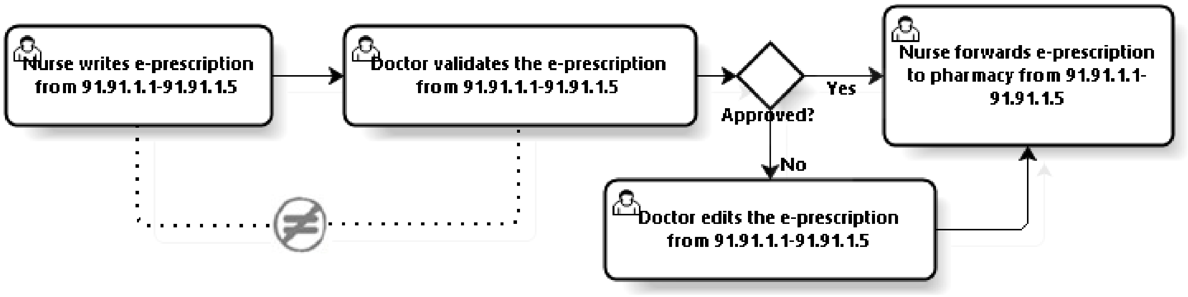

Nurse writes e-prescription from 91.91.1.1-91.91.1.5
Doctor validates the e-prescription from 91.91.1.1-91.91.1.5
Approved?
Yes
No
Nurse forwards e-prescription to pharmacy from 91.91.1.1-91.91.1.5
Doctor edits the e-prescription from 91.91.1.1-91.91.1.5
≠

Figure 5-8 E-prescription review

In the case of outpatients, once the e-prescription is received by the pharmacist, he/she will validate the drugs mentioned in the prescription and if they cannot approve the prescription, the pharmacist may send it back to the doctor for revision. Otherwise, the pharmacist will dispense the drugs to the concerned patient (only after validation), and thereby updates the medical drug database with the remaining drugs.

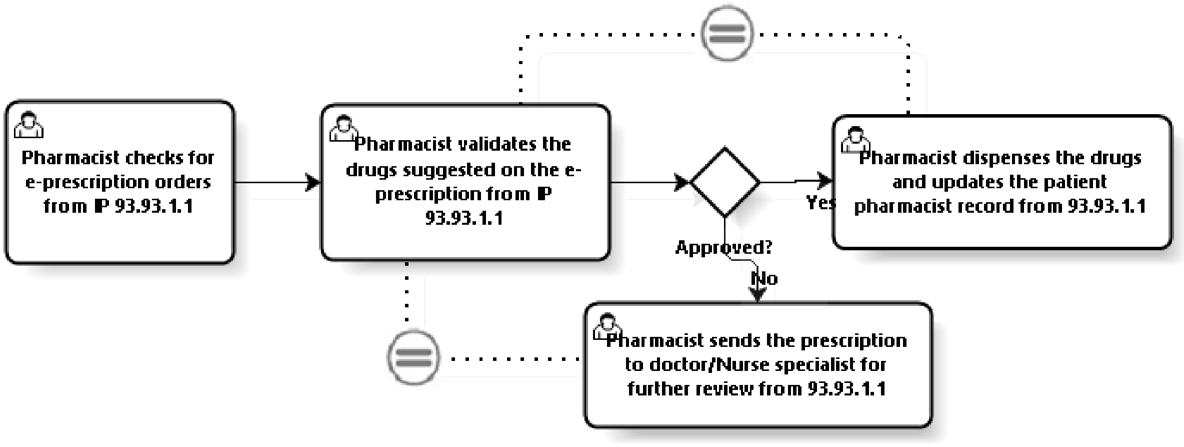

Pharmacist checks for e-prescription orders from IP 93.93.1.1
Pharmacist validates the drugs suggested on the e-prescription from IP 93.93.1.1
Approved?
Yes
No
Pharmacist dispenses the drugs and updates the patient pharmacist record from 93.93.1.1
Pharmacist sends the prescription to doctor/Nurse specialist for further review from 93.93.1.1

Figure 5-9 Pharmacist prescription review

In the case of inpatients, pharmacists will check for e-prescriptions and validates the drugs. If they cannot approve any of the drugs, pharmacist may send it to doctor for further revision. If the pharmacist approves the drugs, he/she will issue to drugs and updates the pharmacy notes and medication storage database of drug issuance. A nurse will dispatch the drugs to the inpatient wards and updates the nurse notes. The bedside nurse will analyze the patient behavior and reactions, and will administer the drugs followed by and update into the nurse notes. Bedside nurse will then save the nurse notes for review by the doctor following which will be entered into the patient clinical record. This scenario can be seen in the below Figure 5-10.

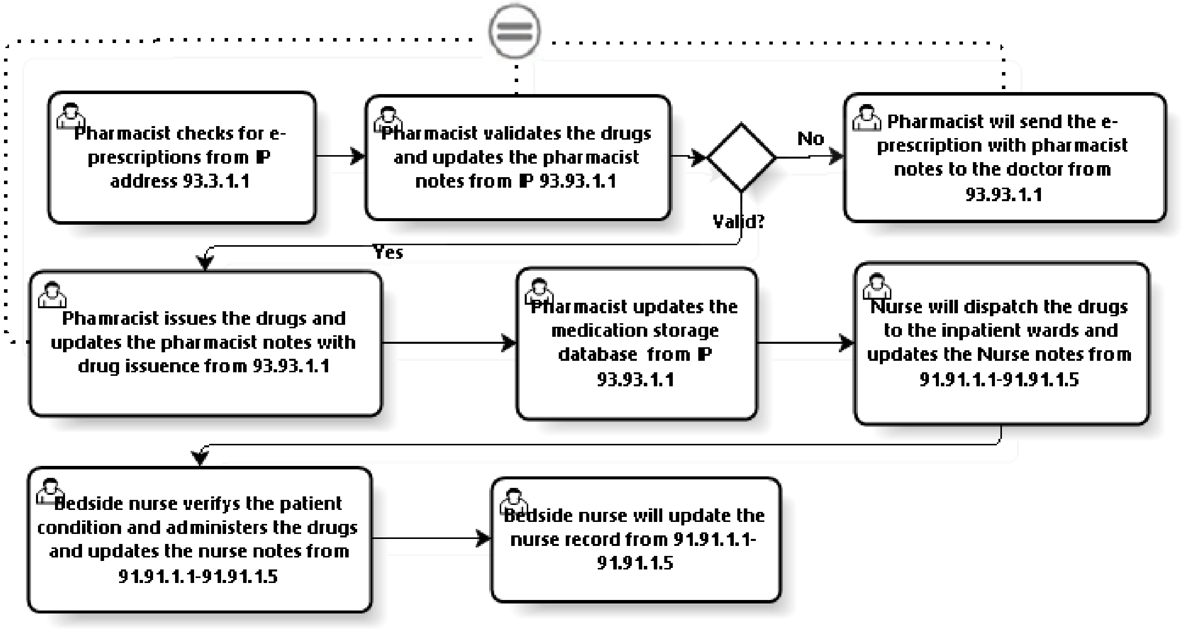

Pharmacist checks for e-prescriptions from IP address 93.3.1.1
Pharmacist validates the drugs and updates the pharmacist notes from IP 93.93.1.1
Valid?
No
Pharmacist wil send the e-prescription with pharmacist notes to the doctor from 93.93.1.1
Yes
Phamracist issues the drugs and updates the pharmacist notes with drug issuence from 93.93.1.1
Pharmacist updates the medication storage database from IP 93.93.1.1
Nurse will dispatch the drugs to the inpatient wards and updates the Nurse notes from 91.91.1.1-91.91.1.5
Bedside nurse verifys the patient condition and administers the drugs and updates the nurse notes from 91.91.1.1-91.91.1.5
Bedside nurse will update the nurse record from 91.91.1.1-91.91.1.5

Figure 5-10 Inpatient drug dispensing

If a patient wants to get a refill of his/her e-prescription, the patient requests the doctor for a refill. Doctor will validate the request, and based on the necessity the doctor may approve or reject the e-prescription. If the request is approved, the renewal notification will be sent to the pharmacist. Upon reciept of the renewal, pharmacist will validate the e-prescription and if the pharmacist finds any invalid information, he/she will send the eprescription back to doctor for further validation, otherwise will dispense the drugs.

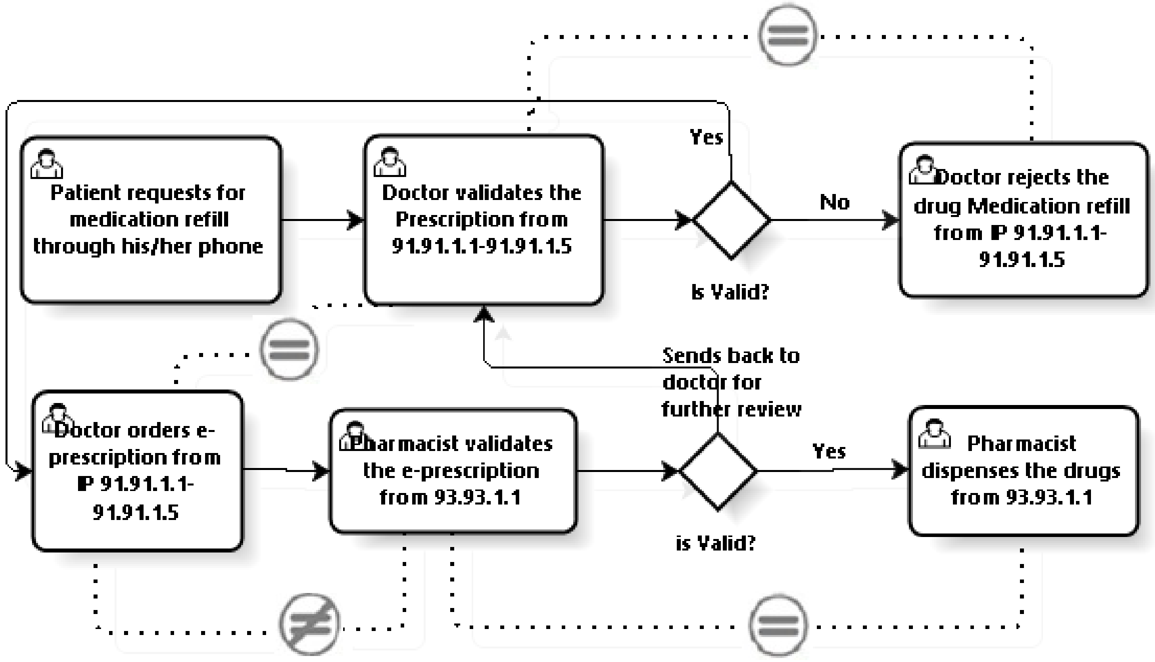

Patient requests for medication refill through his/her phone
Doctor validates the Prescription from 91.91.1.1-91.91.1.5
Yes
No
Is Valid?
Doctor rejects the drug Medication refill from IP 91.91.1.1-91.91.1.5
Sends back to doctor for further review
Doctor orders e-prescription from IP 91.91.1.1-91.91.1.5
Pharmacist validates the e-prescription from 93.93.1.1
Yes
is Valid?
Pharmacist dispenses the drugs from 93.93.1.1

Figure 5-11 Medication refill

In case of an emergency, the emergency response team (ERT) will reach the accident location and will share its location information to the hospital. ERT performs initial analysis on the patient and records the patient behavior into a clinical record. If required, ERT may request for additional information from the hospital. And once ERT reaches the hospital, ERT doctor will conduct an analysis and record the events into the clinical record based on which the doctor may recommend for admission, surgery, or other medications.

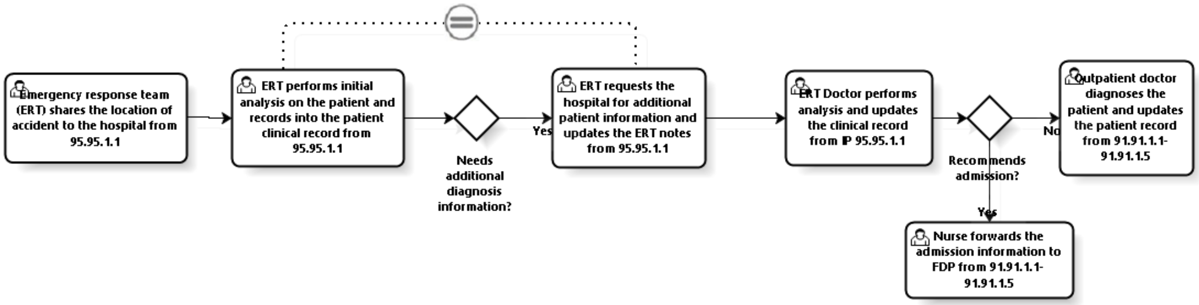

Emergency response team (ERT) shares the location of accident to the hospital from 95.95.1.1
ERT performs initial analysis on the patient and records into the patient clinical record from 95.95.1.1
Needs additional diagnosis information?
Yes
ERT requests the hospital for additional patient information and updates the ERT notes from 95.95.1.1
ERT Doctor performs analysis and updates the clinical record from IP 95.95.1.1
Recommends admission?
No
Outpatient doctor diagnoses the patient and updates the patient record from 91.91.1.1-91.91.1.5
Yes
Nurse forwards the admission information to FDP from 91.91.1.1-91.91.1.5

Figure 5-12 Emergency Response Team

5.2 Attribute based access control Policies

The following is a subset of the access control rules in the case study: Some of the ABAC policies for the proposed case study may include:

PPrac: Any user with the position Practitioner/SupervisorPractitioner can read/write a Patient record from 91.91.1.1-91.91.1.5

SARE: Target is Subject: Practitioner, Resource: Patient Record, Action: read/write, Environment: 91.91.1.1-91.91.1.5

PN: A Nurse can read patient record only from internet network (e.g., IP range: 91.91.1.1-91.91.1.5)

SARE: Target is Subject: Nurse, Resource: patient record, Action: access, Environment: 91.91.1.1-91.91.1.5.

More of these policies can be found in Appendix [2]

5.3 WSO2

WSO2 identity server is an open source application, which hosts a web service and a web application. It offers a user interface with multiple features where a user can log into the system and create, edit, read, or delete an access control policy. A user can login as an administrator to create and manage access control policies which can be tested in another user interface in which the user can send a request in terms of SARE attributes. For example, if a practitioner tries to access a patient health record, the practitioner's credentials are validated against the access control policy file which is located on a XACML engine in the identity server, and after validation the engine will permit/deny the user's request. A detailed demonstration of the WSO2 identity server with an example is discussed in the section 5.4.

5.4 Demonstrating the case study using WSO2

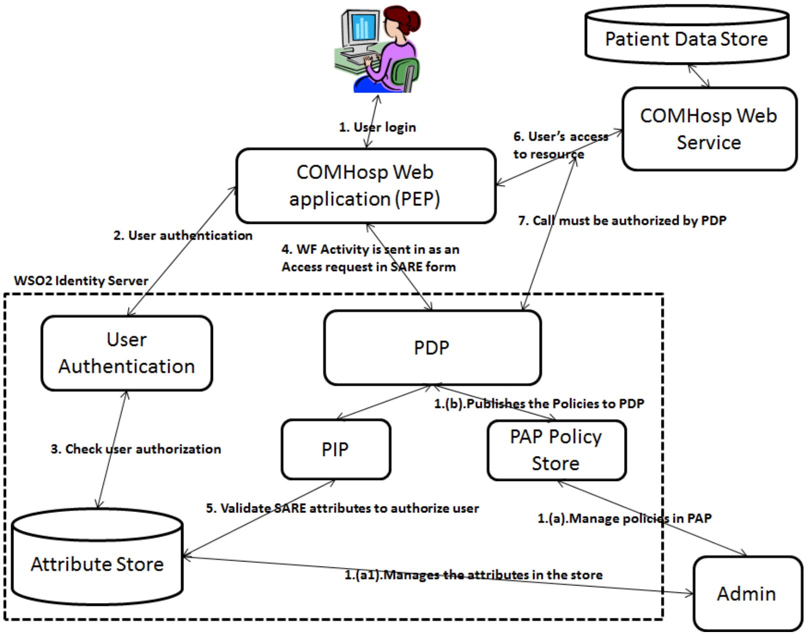

Patient Data Store
COMHosp Web Service
1. User login
6. User's access to resource
COMHosp Web application (PEP)
7. Call must be authorized by PDP
2. User authentication
4. WF Activity is sent in as an Access request in SARE form
WSO2 Identity Server
User Authentication
PDP
1.(b).Publishes the Policies to PDP
3. Check user authorization
PIP
PAP Policy Store
5. Validate SARE attributes to authorize user
1.(a).Manage policies in PAP
Attribute Store
1.(a1).Manages the attributes in the store
Admin

Figure 5-13 Demonstrating the approach using WSO2

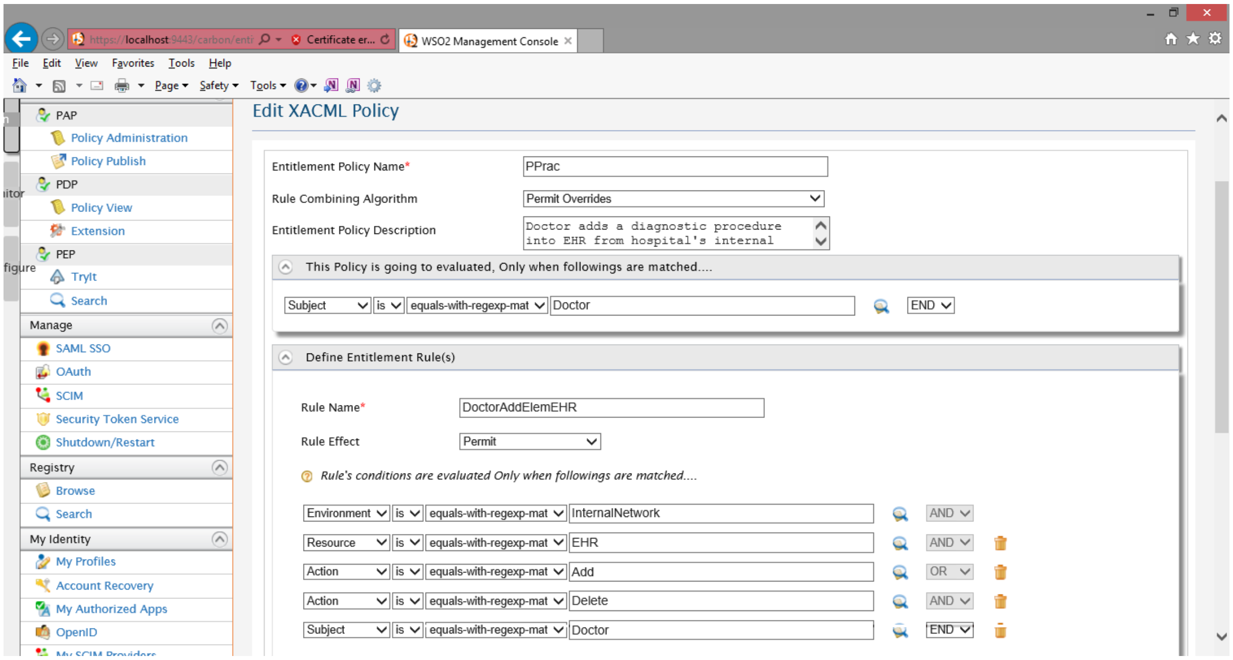
The process of validation of policies on workflow activities is shown in the above Figure 5-13. To validate the model, we have used case study analysis in which we've used Business Process Management Notation (BPMN) to develop the workflow and WSO2 Identity Server to administer the ABAC policies. The steps involved in applying these two for our integrated model are discussed below: ABAC policies written in plain English are entered into the WSO2 Identity server's Policy Administration Point (PAP) through their admin interface, and workflow is developed using BPMN. Once we have policies in WSO2 Identity Server and the workflow, each of the workflow activities is individually identified. Each of the identified workflow activities are extracted from the workflow and the SARE elements are derived from each activity. A request is formulated from the workflow activity's SARE elements, and that request is sent to the WSO2 Identity server's Policy Decision Point. Once the request is sent, WSO2 Identity server will evaluate the request. Based on the policies specification, the server will generate a response containing either ‘permit’ or ‘denied’ or ‘indeterminate’.

- A ‘permit’ response will denote that the intended actions by the subject under certain environmental conditions are allowed.
- A ‘deny’ response will denote that the intended actions by the subject under certain environmental conditions are not allowed.
- An ‘indeterminate’ response will denote that the activity is not fully covered by the policies defined.

To accommodate complex ABAC policies, more coding is required to modify and accommodate more complex WSO2 features. With some coding into WSO2, it is possible to identify which policies are being enforced on the request (workflow activity).

5.5 WSO2 in action

Here, we’ll demonstrate the proposed research methodology using WSO2 Identity server. If we consider activity A1: A Practitioner adds diagnostic procedure into EHR from hospital’s internal network, we send this as a request through WSO2 Identity server (in Figure 5-13). Once the request is evaluated, we can see the result ‘permit’ in the below Figure 5-15.



Edit XACML Policy

Entitlement Policy Name*	PPrac
Rule Combining Algorithm	Permit Overrides
Entitlement Policy Description	Doctor adds a diagnostic procedure into EHR from hospital's internal

This Policy is going to evaluated, Only when followings are matched....

Subject	is	equals-with-regexp-mat	Doctor	END

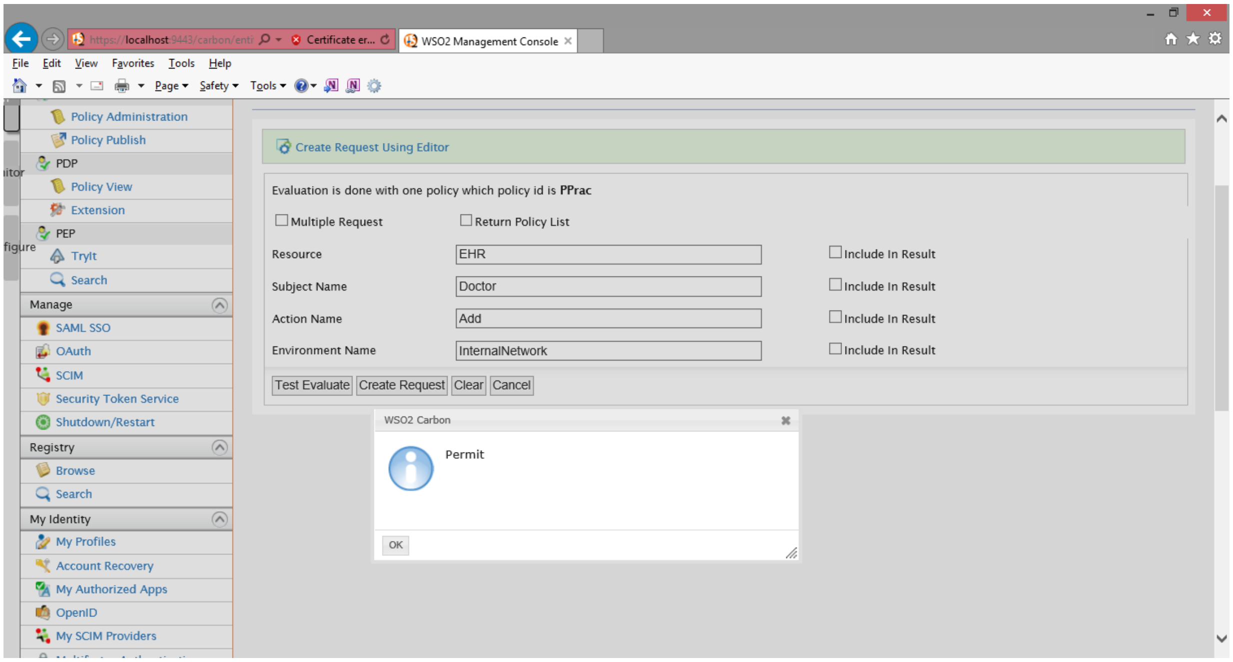
Define Entitlement Rule(s)

Rule Name*	DoctorAddElemEHR
Rule Effect	Permit

Rule's conditions are evaluated Only when followings are matched....

Environment	is	equals-with-regexp-mat	InternalNetwork	AND
Resource	is	equals-with-regexp-mat	EHR	AND
Action	is	equals-with-regexp-mat	Add	OR
Action	is	equals-with-regexp-mat	Delete	AND
Subject	is	equals-with-regexp-mat	Doctor	END

Figure 5-14 PPrac Policy in WSO2



Create Request Using Editor

Evaluation is done with one policy which policy id is **PPrac**

□ Multiple Request □ Return Policy List

Resource	EHR	□ Include In Result
Subject Name	Doctor	□ Include In Result
Action Name	Add	□ Include In Result
Environment Name	InternalNetwork	□ Include In Result

Test Evaluate | Create Request | Clear | Cancel

WSO2 Carbon

Permit

OK

Figure 5-15 Policy PPrac in action

5.6 Testing Results

All of the available ABAC policies are: PPrac, PPR, PFDP, PPayR1, PPayR2, and PN. The activity analysis helps in understanding that not all of the policies are being used across the activities of the workflow, and can be seen in Table 5-2. The policies (y) that are not being used at least once include: PN. The policies that are applicable on each activity of the workflow can be found in Table 5-1 and the policies that are being used (not used) on the workflow activities can be found in Table 5-2.

Table 5-1 Policies applicable on each activity of the workflow

Activity Name	Policy applicable (Yes/ N0/ Partial)?	If applicable, Policy Name
A1	Yes	PPrac, PPR, PIN1
A2	Yes	PPrac, PPR, PIN1
A3	Yes	PFDP, PPayR1, PPayR2, PFDD1
A4	Partial	PPayR1, PPayR2, PFDD1
A5	Partial	PPayR1, PPayR2

Table 5-2 Policies that are being used (or not used) on the workflow activities

Policy	Policy in use (Y/N)?
PPrac	Y
PPR	Y
PFDP	Y
PN	N
PPayR1	Y
PPayR2	Y

PIN1	Y
PFDD1	Y

5.7 GUIDELINES FOR HOSPITALS TO IMPLEMENT THE APPROACH

Purpose

This guide contains the necessary specifications for the integration of workflows and access control policies in a healthcare setting. An integrated methodology will allow efficient and accurate specification and modification of access control policies and workflow activities while reducing the burden on dealing with access control and workflow activities separately. This guide defines how healthcare organizations can define attribute based access control policies; define the implementation of XACML through WSO2; defining and creating a policy administration point; defining the method to extract workflow activities from workflows.

Audience

The intended audiences for this guide are hospital personnel who intend to improve their current access control mechanisms by making environment element their organizational priority. This guide helps in integrating workflow activities and ABAC policies through WSO2 Identity server.

Use case scenarios

Multiple use case scenarios on how to apply the integrated approach through WSO2 and ABAC policies is mentioned in section 5.1.

Steps to implement the proposed approach

Assess current practice

Some of the elements to assess current practice at a healthcare facility may include: (a) performing an analysis on both workflows and access control policies in the organization; (b) performing an analysis on current type of access control mechanism (c) validating the clinical workflows to check if they are developed based on SARE elements; (d) are access control policies and workflow activities using same SARE elements; (e) does the organization considers environment element as a crucial element.

Setting the goals

After assessing the current practice, healthcare organization should set clear goals. To implement the proposed methodology, organization is recommended to set and pursue 'specific' 'measurable' 'attainable' 'relevant' and 'time bound' (SMART) goals (Robert, 2005). The healthcare organization is recommended to make the 'environment' element an organizational priority. Some of the goals to be set by the healthcare organization include: (a) identifying and developing SARE elements across the organization, (b) developing workflow activities in terms of SARE elements, (c) developing ABAC policies in terms of SARE elements, (d) developing policy to use same SARE elements across the development teams, (e) making the environment element an organizational priority, (f) providing training and support to the workflow and policy developer team.

Planning the process

Once the goals are identified, the organizational executive team should develop a plan to implement the methodology and should identify the necessary steps to analyze the process of implementation.

Forming an expert team to identify SARE elements which can be used across the workflow development team and access control developing team

Healthcare organization should identify an expert team to analyze current working process and identify the required SARE attributes so that the same SARE attributes can be used across the teams that develop workflows and access control policies. The purpose of this team is to define the SARE elements to be used commonly by both workflow and the ABAC policies. This team is recommended to have regular meetings to review and add/modify/delete the existing SARE attributes.

Creating a policy statement for SARE elements usage across the organization

To follow uniform standards of data usage across various electronic platforms, healthcare organization should make a policy statement about SARE elements across the organizational development teams. The organization should also make the policy statement available to the development teams and should work on making the teams to understand the statement.

Training on developing workflow activities in terms of SARE elements

Proper training to the workflow developers on how to develop workflow activities in terms of SARE elements should be given. It is an important task to develop workflow activities in terms of SARE elements as these are the elements which would help in integrating the workflow activities with the ABAC policies.

Plan to implement ABAC through WSO2 identity server

As WSO2 identity server is an open source application and accommodates healthcare requirements, WSO2 identity server has been used to implement the approach in this dissertation. As the WSO2 identity server is an open source application and with proper coding knowledge, it can be easily modified based on the organization’s requirement. It is recommended by the author to the organization to plan to implement ABAC policies through WSO2 identity server.

Employing an ABAC policy and WSO2 Identity server expert

To implement and develop ABAC policies through WSO2, the developer team should have sufficient knowledge about the concept, which requires having an ABAC policy and WSO2 Identity server expert on team. ABAC policies should be developed from the SARE elements recognized by the expert team. The policy expert can provide required support and guidance to the policy developing team.

Training on ABAC and WSO2 Identity server

In order to implement ABAC through WSO2 Identity server, the team should have proper training about ABAC concepts, how to implement WSO2, how WSO2 works, how to write ABAC policies in WSO2, how to validate requests through WSO2, and also how to edit the WSO2 identity server.

6 SUMMARY

This dissertation addresses crucial problems in two different scenarios, (i) for healthcare organizations that are using workflows and role to task assignments, we have developed an algorithm to analyze workflow instances for obstructions due to static and dynamic authorization policies that allow organizations to properly assign users to tasks without the policies causing obstructions; and (ii) for future healthcare organizations which accommodate environment into workflow activities and access control policies, we have addressed a crucial objective: integrating the workflows and access control policies; And our integrated methodology can identify workflow activities that are not being protected by access control policies. They can further be improved by modifying the workflow activities and access control policies through integrating workflow activities and attribute based access control policies using SARE (Subject, Action, Resource, and environment) elements.

Our main contributions include: (i) an algorithm to test the workflow instances for any obstructions with respect to policies, (ii) proposing ‘environment’ as a crucial element in a workflow activity along with Subject, Action, and Resource, (iii) a methodology to integrate workflow activities and access control policies, using this methodology to address below questions: (a) Whether there are any workflow activities that are not covered by the access control policies, (b) Whether there are any unused access control policies, (c) How many policies are covering each activity, and (d) Whether the workflow activities and the access control policies need to be modified or refined, (e) Demonstrating the methodology with WSO2 Identity Server.

As a part of our future work we have plans to work on: (i) extend the case study to perform in-depth testing and analysis with multiple healthcare scenarios, (ii) modifications to WSO2 identity server are needed to accommodate extended policy sets, (iii) modifications to WSO2 are needed to view the policies that are effecting a workflow activity, (iv) identify any policy related conflict issues, (v) perform in-depth analysis of the integrated methodology.

7 REFERENCES

Aalst, W. M. P. van der. (2004). Business process management: A personal view. *Business Process Management Journal, 10*(2), 135–139.

Akinyele, J. A., Pagano, M. W., Green, M. D., Lehmann, C. U., Peterson, Z. N. J., & Rubin, A. D. (2011). Securing electronic medical records using attribute-based encryption on mobile devices. *Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices SPSM 11*, 75–86. doi:10.1145/2046614.2046628

Alhaqbani, B., & Fidge, C. (2007). Access control requirements for processing electronic health records. *Proceedings of the 2007 International Conference on Business Process Management BPM'07*, 371–382. doi:10.1007/978-3-540-78238-4_38

Basin, D., Burri, S. J., & Karjoth, G. (2011). Obstruction-free authorization enforcement: Aligning security with business objectives. In *Proceedings - IEEE Computer Security Foundations Symposium* (pp. 99–113).

Bertino, E., Ferrari, E., & Atluri, V. (1999). The specification and enforcement of authorization constraints in workflow management systems. *ACM Transactions on Information and System Security*.

Campbell, E. M., Sittig, D. F., Ash, J. S., Guappone, K. P., & Dykstra, R. H. (2006). Types of unintended consequences related to computerized provider order entry. *Journal of the American Medical Informatics Association, 13*(5), 547–556.

Chaari, S., Biennier, F., Amar, C. Ben, & Favrel, J. (2004). An authorization and access control model for workflow. *First International Symposium on Control, Communications and Signal Processing, 2004.* doi:10.1109/ISCCSP.2004.1296239

Cole, K. (2002). HIPAA Compliance: Role Based Access Control. *Global Information Assurance Certification Paper - SANS Institute*.

Cole, R., Purao, S., Rossi, M., & Sein, M. K. (2005). Being proactive: where action research meets design research. In *Proceedings of the TwentySixth International Conference on Information Systems* (pp. 1–21). Association for Information Systems.

Eekels, J., & Roozenburg, N. F. M. (1991). A methodological comparison of the structures of scientific research and engineering design: their similarities and differences. *Design Studies*, *12*(4), 197–203. doi:10.1016/0142-694X(91)90031-Q

Ferraiolo, D. F., Sandhu, R., Gavrila, S., Kuhn, D. R., & Chandramouli, R. (2001). Proposed NIST Standard for RBAC. *ACM Transactions on Information and System Security*, *4*(3).

Ferreira, A., Cruz-Correia, R., Antunes, L., & Chadwick, D. (2007). Access control: how can it improve patients' healthcare? *Studies in Health Technology and Informatics*, *127*, 65–76.

Giaglis, G. M. (2001). A taxonomy of business process modeling and information systems modeling techniques. *International Journal of Flexible Manufacturing Systems*, *13*(2), 209–228.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly*, *28*(1), 75–105. doi:10.2307/249422

HHS.Gov. Health Information Privacy - HIPAA Privacy Rule. , U.S. Department of Health & Human Services, HHS.gov (2003).

Hu, V. C., Scarfone, K., & Kuhn, R. NIST Special Publication 800-162 DRAFT - FINAL Guide to Attribute Based Access Control (ABAC) Definition and Considerations NIST Special Publication 800-162 DRAFT - FINAL Guide to Attribute Based Access Control (ABAC) Definition and Considerations (2013).

Identifier, D. (2005). eXtensible Access Control Markup Language OASIS Standard , 1 Feb 2005. *Oasis Standard*, *200502*(February), 23.

Janssen, C. (2014). Security Breach. *TechoPedia*.

Lakkaraju, S. (2013). Context Aware Mobile Knowledge Management System in Healthcare. In *MWAIS* (pp. 1–10).

Lakkaraju, S., & Lakkaraju, S. K. (2011). Mobile Device Application in Healthcare. In *Cases on Healthcare Information Technology for Patient Care Management* (p. 22).

Lakkaraju, S., & Moran, M. (2011). A Framework to Investigate the Role of Mobile Technology in the Healthcare Organizations. *MWAIS*.

Lakkaraju, S., & Xu, D. (2014). Integrated Modeling and Analysis of Attribute based Access Control Policies and Workflows in Healthcare. In *Proc. of the 1st International Conference on Trustworthy Systems and Their Applications (TSA'14), Taiwan* (p. 8).

Le, X. H., Doll, T., Barbosu, M., Luque, A., & Wang, D. (2012). An enhancement of the Role-Based Access Control model to facilitate information access management in context of team collaboration and workflow. *Journal of Biomedical Informatics*, *45*(6), 1084–1107.

Leyva, C. (2014). Hitech Act Summary. *HIPAA Survival Guide*. Retrieved from http://www.hipaasurvivalguide.com/hitech-act-summary.php

Ligatti, J., Bauer, L., & Walker, D. (2005). Edit automata: Enforcement mechanisms for run-time security policies. *International Journal of Information Security*, *4*(1-2), 2–16.

Lowe, G. (1996). Some new attacks upon security protocols. In *9th IEEE Computer Security Foundations Workshop* (pp. 162–169).

Lu, Y., Zhang, L., & Sun, J. (2009). Task-activity based access control for process collaboration environments. *Computers in Industry*, *60*(6), 403–415.

March, S. T., & Smith, G. F. (1995). Design and natural science research on information technology. *Decision Support Systems*, *15*(4), 251–266. doi:10.1016/0167-9236(94)00041-2

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, *24*(3), 45–77. doi:10.2753/MIS0742-1222240302

Robert, B. (2005). Use S.M.A.R.T Goals to Launch Management by Objectives Plan. *TechRepublic*. Retrieved from http://www.techrepublic.com/article/use-smart-goals-to-launch-management-by-objectives-plan/

Roscoe, A. W. (2005). *The Theory and Practice of Concurrency. Theory and Practice* (Vol. 216). Prentice Hall.

Russello, G., Dong, C. D. C., & Dulay, N. (2008). A Workflow-Based Access Control Framework for e-Health Applications. *22nd International Conference on Advanced Information Networking and Applications - Workshops (aina Workshops 2008)*. doi:10.1109/WAINA.2008.131

Sandhu, R. (1988). Transaction control expressions for separation of duties. *[Proceedings 1988] Fourth Aerospace Computer Security Applications*.

Schneider, F. B. (2003). Enforceable security policies. *Foundations of Intrusion Tolerant Systems, 2003 [Organically Assured and Survivable Information Systems]*.

Siriwardena, P. (2013). Understanding HIPAA. *Facilelogin*. Retrieved from http://blog.facilelogin.com/2013/02/understanding-hipaa.html

Sittig, D. F., Krall, M., Kaalaas-Sittig, J., & Ash, J. S. (2005). Emotional Aspects of Computer-based Provider Order Entry: A Qualitative Study. *Journal of the American Medical Informatics Association*, *12*(5), 561–567.

Spear, N., Malladi, S., & Lakkaraju, S. (2012). Analyzing Workflows in Business Processes for Obstructions Due to Authorization Policies. In *45th Hawaii International Conference on System Sciences* (pp. 5340–5349).

Tan, K., Crampton, J., & Gunter, C. A. (2004). The consistency of task-based authorization constraints in workflow. *Proceedings. 17th IEEE Computer Security Foundations Workshop, 2004.*

United States Department of Defense DoD. Trusted Computer System Evaluation Criteria (1985).

VAN DER AALST, W. M. P. (1998). THE APPLICATION OF PETRI NETS TO WORKFLOW MANAGEMENT. *Journal of Circuits, Systems and Computers*.

Welch, J. (2014). Improving Clinical Workflow: An Example from the Emergency Department. *Health Catalyst*. Retrieved from https://www.healthcatalyst.com/improve-clinical-workflow-emergency-department

Wolter, C., Schaad, A., & Meinel, C. (2008). Task-based entailment constraints for basic workflow patterns. In *Proceedings of the 13th ACM symposium on Access control models and technologies - SACMAT '08* (pp. 51–60). ACM Press.

XiangPeng, Z., Cerone, A., & Krishnan, P. (2006). Verifying BPEL Workflows Under Authorization Constraints. *Business Process Management, 4th International Conference, BPM, 4102*, 439–444.

Xu, D., & Zhang, Y. (2014). Specification and Analysis of Attribute-Based Access Control Policies: An Overview. *International Workshop on Information Assurance, in Conjunction with the 8th International Conference on Software Security and Reliability (SERE'14).*

Yuan, E., & Tong, J. (2005). Attributed based access control (ABAC) for Web services. *IEEE International Conference on Web Services (ICWS'05)*.

Zhang, G., & Liu, J. (2011). A Model of Workflow-oriented Attributed Based Access Control. *International Journal of Computer Network and Information Security*, *1*(February), 47–53.

8 APPENDICES

8.1 Appendix [1] policy in xacml (wso2 Identity server)

Any doctor can read and write into patient health record (PHR) only for his/her designated patients from internal network

```
        <Rule RuleId="urn:oasis:names:tc:xacml:3.0:example:ruleid:1" Effect="Permit">
                <Description>Any doctor can read and write into patient health record (PHR) only for his/her
designated patients from internal network </Description>
         <Target>
         <AnyOf><AllOf><Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">doctor</AttributeValue>
                    <AttributeDesignator MustBePresent="false"
                  Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
                     AttributeId="urn:oasis:names:tc:xacml:3.0:example:attribute:role"
                     DataType="http://www.w3.org/2001/XMLSchema#string"/></Match></AllOf></AnyOf>
        <AnyOf><AllOf><Match MatchId="urn:oasis:names:tc:xacml:3.0:function:xpath-node-match">
                    <AttributeValue DataType="urn:oasis:names:tc:xacml:3.0:data-type:xpathExpression"
             XPathCategory="urn:oasis:names:tc:xacml:3.0:attribute-category:resource">Patienthealth
record</AttributeValue><AttributeDesignator MustBePresent="false"
                    Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
                     AttributeId="urn:oasis:names:tc:xacml:3.0:content-selector"
                    DataType="urn:oasis:names:tc:xacml:3.0:data-type:xpathExpression"/>
                  </Match></AllOf></AnyOf>
        <AnyOf><AllOf><Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">read</AttributeValue>
                    <AttributeDesignator MustBePresent="false"
                     Category="urn:oasis:names:tc:xacml:3.0:attribute-category:action"
                     AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
                     DataType="http://www.w3.org/2001/XMLSchema#string"/>
                  </Match></AllOf></AnyOf>
        <AnyOf><AllOf><Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">write</AttributeValue>
                    <AttributeDesignator MustBePresent="false"
                     Category="urn:oasis:names:tc:xacml:3.0:attribute-category:action"
```

```
                      AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
                      DataType="http://www.w3.org/2001/XMLSchema#string"/>
                    </Match> </AllOf></AnyOf>
          <AnyOf><AllOf><Match MatchId="urn:oasis:names:tc:xacml:3.0:function:xpath-node-match">
                      <AttributeValue DataType="urn:oasis:names:tc:xacml:3.0:data-type:xpathExpression"
              XPathCategory="urn:oasis:names:tc:xacml:3.0:attribute-category:environment">internal
network</AttributeValue><AttributeDesignator MustBePresent="false"
                      Category="urn:oasis:names:tc:xacml:3.0:attribute-category:environment"
                       AttributeId="urn:oasis:names:tc:xacml:3.0:content-selector"
                      DataType="urn:oasis:names:tc:xacml:3.0:data-type:xpathExpression"/>
                    </Match></AllOf></AnyOf>
          </Target>
          <Condition>
                  <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                   <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
                    <AttributeDesignator MustBePresent="false"
                    Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
                  AttributeId="urn:oasis:names:tc:xacml:3.0:example: attribute:physician-id"
                     DataType="http://www.w3.org/2001/XMLSchema#string"/></Apply>
                   <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
                    <AttributeSelector MustBePresent="false"
                     Category="urn:oasis:names:tc:xacml:3.0:attribute-
category:resource"Path="/doctor/designatedpatient/"
                     DataType="http://www.w3.org/2001/XMLSchema#string"/>
                  </Apply>
          </Condition>
          </Rule>
```

8.2 Appendix [2] ABAC Policies

The following is a subset of the access control rules in the case study:

PPrac: Any user with the position Practitioner/SupervisorPractitioner and an employee of the Organization, and are assigned to the patient, can read/write a Patient record from 91.91.1.1-91.91.1.5

SARE: Target is Subject: Practitioner, Resource: Patient Record, Action: read/write, Environment: 91.91.1.1-91.91.1.5

PN: A Nurse who is working under a practitioner can access patient record from internet network (e.g., IP range: 91.91.1.1-91.91.1.5)

SARE: Target is Subject: Nurse, Resource: patient record, Action: access, Environment: 91.91.1.1-91.91.1.5.

PFDP: Any user with Position: FrontDeskPerson from Department Front Desk with UserID Receptionist can 'access' the 'insurance information' of a patient and thereby 'validate' the Insurance of a patient and add/update it to the payment record, from 92.92.1.1.

SARE: Target is Subject: FrontDeskPerson, Resource: Payment Record, Action: Add/Update, Environment: 92.92.1.1

PPR: Patient Record cannot be accessed by any user with Position: FrontDeskPerson from the IP 92.92.1.1.

If tried to access, email to the Supervisor 'FrontdeskPerson with name X has tried to access Clinical Record X from ip address 92.92.1.1.

SARE: Target is Resource: 'Patient Record', Subject: FrontdeskPerson, action: access, Environment: 92.92.1.1.

PPayR1: Any user with Designation: Patient whose age is greater than 16 can read his/her own Payment Record from internal network or from his/her mobile device at any time.

SARE: Target is Resource: Payment Record, Subject: Patient, Action: read, Environment: Hospital's internal network or from his/her mobile device

PPayR2: Any user with Designation: guardian or parent can read Payment Record of a Patient whose age is less than 16 from hospital's internal network or from his/her mobile device.

SARE: Target is Resource: Payment Record, Subject: Gaurdian/Parent, Action: read, Environment: hospital's internal network or from his/her mobile device.

PIN1: Practitioner/Nurse/SupervisorPractitioner can only read/write patient record from 91.91.1.1-91.91.1.5.

SARE: Target is Subject: Practitioner or Nurse or SupervisorPractitioner, Resource: Patient Record, Action: read/write, Environment: 91.91.1.1-91.91.1.5

PFDD1: Only FDPerson can be logged into 'FDDesktop', and read PatientInsuranceInformation.

SARE: Target is Subject: FDPerson; Resource: Patient Insurance Information, Action: login, Environment: 92.92.1.1;

PPEHR1: Any user with Position: FDP from Department Front Desk with UserID Receptionist* can read/write Receptionist Record from Patient EHR, but cannot access Clinical Record from 92.92.1.1.

PPEHR2: Any user with Designation: Patient whose age is greater than 16 can access (read) his/her own Patient EHR from a mobile device at any time.

PPEHR3: Any user with Designation: guardian or parent can read Patient EHR of a Patient whose age is less than 16 from from his/her mobile device at any time.

PCR1: Any user with the position Doctor/Nurse and an employee of the Organization, and are assigned to the patient, can read/write a clinical record from 91.91.1.1-91.91.1.5 from any device.

PCR2: Any user with position Nurse/Doctor and any NurseDesignation/DoctorDesignation can read clinical record in the morning shift between 8:00 AM to 6:00 PM and an evening shift between 6:00PM to 8:00AM, from 91.91.1.1-91.91.1.5 from any device.

If the Clinical Record is not in use, it will be closed in 10 minutes.

PCR3: Clinical Record (any part of CR, NR, DR, PR, and LR) cannot be accessed by any user with Position: Receptionist and from department FrontDesk, from any IP address, at any time, from any device.

If tried to access, email to the Supervisor 'Receptionist with name X has tried to access Clinical Record X from ip address x.x.x.x, from device X.

PDR1: Any user with Position: Doctor and is designated with a patient can create a Doctor record at any time from ip address 91.91.1.1-91.91.1.5, from any device.

Condition: Doctor should mention the DoctorDesignation at the time of creation of the record.

PDR2: Any user with Position: Doctor and and is designated with a patient can read/add/write Doctor Notes with doctor Record at any time from ip address 91.91.1.1-91.91.1.5 in the hospital, from any device

PDR3: Any user with Position: Doctor and is designated with a patient can read/add doctor record to the clinical record at any time from ip address 91.91.1.1-91.91.1.5 in the hospital, from any device

PNR1: Any user with Position: Doctor/Nurse can read/write Nurse Record in the morning shift between 8:00 AM to 6:00 PM, and in the evening shift between 6:00PM and 8:00 AM, from

91.91.1.1-91.91.1.5 from any device.

Condition: If the Nurse Record is not in use, it will be closed in 10 minutes.

PNR2: Nurse Record can be read/modified/deleted by any user with position: doctor under whose supervision the nurse is working at any time from hospital ip address 91.91.1.1-91.91.1.5 and from any device.

PPR1: Prescription can be created by any user with position: doctor at any time from hospital ip address 91.91.1.1-91.91.1.5 from any device.

Condition: Add doctor Designation/UserID and department

PPR2: Recommendation for refill can be done by any designated user with position: doctor at any time from hospital ip address 91.91.1.1-91.91.1.5 from any device.

Condition: Add doctor Designation/User ID and department

PPR3: any user with position: Nurse can create a prescription at any time from hospital ip address 91.91.1.1-91.91.1.5 from any device, except for controlled substances.

Condition: If the prescription is for controlled substances, Nurse should get permission from the doctor

PRR1: Receptionist Record can be created by any user with Position: Receptionist and from Department: Frontdesk any time from 92.92.1.1 from any device

PRR2: Receptionist Record cannot be accessed (read, write, create, delete, update) by any user with position: doctor/nurse at any time from any hospital ip address from any device.

FDP asks patient relatives to sign a patient consent form reception

Subject: PatientRelative/PatientFriend Resource: PatientConsentform Action: Sign the form Environment: only from 92.92.1.1 *Condition: Should be a Relative/Friend*

FDP can attach the consent form only from 92.92.1.1 and only between 8 to 5 (in certain time frame).

Subject: FDP Resource: ConsentForm, EHR Action: Add form to Patient EHR Condition 1: should be from 92.92.1.1 Condition 2: should be between 8 AM to 5PM.

FDP assigns a Nurse to the ER patient from 92.92.1.1

Subject: Receptionist Resource: Nurse Action: Assign Environment: only from 92.92.1.1 Condition: should be between 8 AM to 5PM

Nurse assigns a bed to the patient from 91.91.1.1-91.91.1.5

Subject: Nurse Resource: Bed Action: Assign Environment: only from 91.91.1.1-91.91.1.5

Assigned Nurse will conduct initial assessment of the patient's condition, Order basic blood tests only from 91.91.1.1-91.91.1.5

Rule 1: Nurse records patient's condition into the EHR

Subject: Nurse Resource: EHR Action: Record/Edit Environment: only from 91.91.1.1-91.91.1.5

Rule 2: Nurse orders lab tests for patient

Subject: Nurse Resource: Lab test Action: Order Environment: only from 91.91.1.1-91.91.1.5

Condition: should be between 8 –5

Retrieve earlier (if any) patient records/medications/drug reports/ etc. Assigned nurse can retrieve patient records between 8-5 from 91.91.1.1-91.91.1.5

Subject: Nurse Resource: PatientEHR Action: read Environment: only from 91.91.1.1-91.91.1.5

Condition: should be between 8 –5

Add Nurse assessment (Nurse Notes) to the patient record only from 91.91.1.1-91.91.1.5

Subject: Nurse Resource: NurseNotes Action: Add Environment: only from 91.91.1.1-91.91.1.5

Condition: should be between 8 –5

Lab reports should be done by certified professional from 94.94.1.1and should Perform lab tests within 2 hours.

Subject: Lab Professional Resource: Lab report Action: Add Environment: only from 94.94.1.1

Condition: should be between 8 –5 Obligation: should be done within 2 hours.

IP address 91.91.1.1-91.91.1.5 should be accessed only by nurse/doctor

Subject: Doctor, Resource: Record, Action: Access, Environment: 91.91.1.1-91.91.1.5

Condition: Doctor is the only person who can access the ipaddress.

IP address 92.92.1.1 should only be accessed by receptionist/FDP

Subject: FDP, Resource: Record, Action: Access, Environment: 92.92.1.1

Condition: front desk person (FDP) is the only person who can access the ipaddress.

IP address 93.93.1.1 should only be accessed by pharmacist

Subject: Pharmacist, Resource: Prescription, Action: Access, Environment: 93.93.1.1

Condition: Pharmacist is the only person who can access the ipaddress.

IP address 94.94.1.1should only be accessed by lab specialists

Subject: Lab specialists, Resource: LabRecord, Action: Access, Environment: 94.94.1.1

Condition: Lab specialist is the only person who can access the ipaddress.