CATÓLICA
ESCOLA SUPERIOR DE BIOTECNOLOGIA

PORTO

Creation and Evaluation of a Patient Demographics Mismatch Estimator

by

Randy QUINTUS

June 2021

CATÓLICA
ESCOLA SUPERIOR DE BIOTECNOLOGIA
PORTO

Creation and Evaluation of a Patient Demographics Mismatch Estimator

Training Placement Report presented to *Escola Superior de Biotecnologia* of the *Universidade Católica Portuguesa* to fulfill the requirements of Master of Science degree in Biomedical Engineering

by

Randy QUINTUS

Supervisor (Artces Unipessoal, Lda.): Senior Deployment & Service Engineer, Daniel Carvalho

Supervisor (Universidade Católica Portuguesa): Dr. Pedro Miguel de Luís Rodrigues

June 2021

Abstract

Based on the constant technological advancements, modern hospitals have been undergoing massive changes to stay conform with the directives for the fourth industrial revolution. As a result, these hospitals are progressively digitalizing their internal processes. However, with the ever increasing data size due to the digitalization process, it is important ensuring that all the data is processed correctly and that none of its content is treated wrongly.

Especially during the patient's data processing, the system needs to guarantee that the correct information is allocated to the right patient. In fact, patient mismatches can lead to redundant tests, incorrect treatment services, unnecessary hospitalizations, wrong diagnosis as well as a significant increase of additional costs. The worst incorrect patient identification scenarios include: missed cancer, unnecessary mastectomy, hemolytic reactions to transfusion and even death.

This document aims to present the report concerning the curricular internship for the Master's degree in Biomedical Engineering of the Catholic University of Portugal in Porto. The internship was conducted at Sectra, a medical information technology (IT) company which is specialised in the development of Picture Archiving and Communication Systems (PACS), a product used by the radiology departments of modern hospitals.

The main goal of the internship was to create a tool which would be used to prevent possible patient misidentifications during a PACS migration process. Additionally, a research regarding the IT infrastructure of a modern radiology department as well as the related workflow was made. The tool was developed by using *Windows Powershell*, *Excel* and *PostreSQL*. The tests of the tool were made in an IT-laboratory environment containing anonymized patient data.

The mismatch identification tool appeared to work as designed as it was able to fulfill all the requested criterias. Therefore, the tool could be used in a production-like environment where real patient data would be processed. However, it is recommended to add some features to the tool so that it would become even more valuable to its users. Some of the advised features would be the display of the migration duration based on the detected mismatches and an automated fixation of the detected mismatches.

In addition to scripting the tool, one if the biggest challenges during the internship was to fully understand the PACS workflow as well as the image acquisition according to Integrating the Health Enterprises (IHE). In this aspect, several fluxograms were created to demonstrate a deeper understanding of the subjects. Furthermore, a proficient knowledge regarding the messaging standard: Health Level 7 (HL7) and Digital Imaging and Communications in Medicine (DICOM) were required to successfully deliver the tasks.

Keywords: Patient Identification, Radiology Department, PACS, Standards

Resumo

Com base nos constantes avanços tecnológicos, os hospitais modernos têm vindo a passar por grandes mudanças para se manterem em conformidade com as regras da revolução industrial 4.0. Como resultado, os hospitais modernos estão a digitalizar progressivamente os seus processos internos. No entanto, com o aumento progressivo do tamanho dos dados procesados devido ao processo de digitalização, é importante garantir que todos os dados sejam processados corretamente e que nenhum dos seus conteúdos seja tratado incorretamente.Especialmente durante o processamento de dados de pacientes, o sistema precisa de garantir que as informações corretas são alocadas ao paciente certo. De facto a identificação incorreta dos dados de pacientes pode levar a testes redundantes, tratamentos incorretos, hospitalizações desnecessárias, diagnósticos errados, bem como um aumento significativo de custos adicionais. Aquando da identificação incorreta dos dados de paciente, os piores cenários podem incluir: cancro não-identificado, mastectomia desnecessária, reações hemolíticas à transfusão de sangue e até a morte.

O presente documento tem como objetivo apresentar o relatório relativo ao estágio curricular do Mestrado em Engenharia Biomédica da Universidade Católica Portuguesa do Porto. O estágio foi realizado na Sectra, empresa de tecnologia da informação médica especializada no desenvolvimento de Sistemas de Arquivamento e Comunicação de Imagens (PACS), produto utilizado pelos departamentos de radiologia de hospitais modernos.

O objetivo principal do estágio era criar uma ferramenta que pudesse ser usada para prevenir possíveis erros de identificação de pacientes durante um processo de migração do PACS. Além disso, foi feita uma pesquisa acerca da infraestrutura de tecnologia da informação de um moderno departamento de radiologia, bem como do fluxo de trabalho relacionado.

A ferramenta foi desenvolvida em *Windows Powershell*, *Excel* e *PostreSQL* e os testes da ferramenta foram feitos em ambiente de laboratório de informática contendo dados anónimos de pacientes.

A ferramenta de identificação de incompatibilidades aparentou funcionar conforme projetado, pois era capaz de atender a todos os critérios solicitados. Portanto, a ferramenta poderia ser usada num ambiente de produção, onde os dados reais de pacientes seriam processados. No entanto, é recomendável adicionar alguns recursos à ferramenta para que ela se torne ainda mais valiosa para os seus utilizadores. Alguns dos recursos recomendados seriam a exibição da duração da migração com base nas incompatibilidades detectadas e uma fixação automatizada das incompatibilidades detectadas.

Além da criação do script da ferramenta, um dos maiores desafios encontrados durante o estágio foi compreender completamente o fluxo de trabalho do PACS, bem como a aquisição de imagens de acordo com a *Integrating the Health Enterprises (IHE).* Por essa razão, diversos fluxogramas foram criados para demonstrar um conhecimento mais aprofundado destes assuntos. Adicionalmente, foi necessário adquirir um conhecimento proficiente acerca do padrão de mensagens: *Health Level 7 (HL7)* e *Digital Imaging and Communications in Medicine (DICOM)*.

Keywords: Identificação dos pacientes, Departamento de Radiologia, *PACS*, Padrões

Acknowledgements

I would like to thank Sectra Iberia for giving me the opportunity to do my dissertation on such a challenging but interesting field. Thank you Sectra for the great work environment and for your support. A special thank you to my supervisors Daniel Carvalho and Prof. Pedro Rodrigues for the continuous help and feedback, especially during this unusual time.

I would also like to thank my family and friends for always supporting me and for inspiring me to do my best.

Table of contents

1. Introduction 1
1.1 Motivation 1
1.1.1 The challenge of patient matching: 1
1.1.2 Consequences of patient mismatches: 2
1.1.3. Cause of patient mismatches: 4
1.2 Purpose and main goal of the task 5
1.3. Dissertation Structure 5
2. Structure and Schedule of the Internship 6
3. Background Theory 9
3.1 Digital Imaging and Communications in Medicine (DICOM) 9
3.1.1 Definition 9
3.1.2 DICOM Vocabulary 10
3.1.3 DICOM Information Model 10
3.1.7 Putting all together 16
3.2 The Health Level 7 Standard 18
3.2.1 Definition: 18
3.2.2 Advantages of using a messaging standard: 19
3.2.3 Versions of the HL7 Standard: 20
3.2.4 Application of HL7: 21
3.3 Radiology Department 23
3.4 Hospital Information System (HIS) 23
3.5 Radiology Information System (RIS) 25
3.6 Picture Archiving and Communication System Components and Workflow 27
3.6.1 PACS Components 27
3.6.2 Functionality of PACS 28
3.6.3 A Generic PACS Workflow 29
3.7 Migration of Medical Images and Related Data 31

3.7.1 Definition 31
3.7.2 Purpose of Data Migration 31
3.7.3 Data Migration Technique 32
3.8 Integrating the Health Enterprises (IHE) 33
3.9 Sectra 36
4. Project activities 37
4.1 Examination Mismatch 37
4.1.1 How images are imported into PACS: 38
4.1.2 DICOM image mismatch procedure: 39
4.2 Methodology of the project 42
4.2.1 Structured Query Language (SQL) 42
4.2.2 Powershell 44
4.2.3 Image acquisition workflow according to IHE standards: 45
4.2.4 Creating the mismatch estimation tool: 50
5. Conclusion and Future Work 56
5.1 Results 56
5.2 Future Work 57
5.3 Conclusion 57
Appendix 59
References 79

List of Figures

Figure 2.1: Gantt Chart - Internship 6
Figure 3.1: From real data to DICOM IODS 10
Figure 3.2: DICOM Attribute (Data Element) Structure 11
Figure 3.3: DICOM Application Entitites 12
Figure 3.4: Structure Of DICOM DIMSE 13
Figure 3.5: Example On How DICOM Application (AES) Communicate 13
Figure 3.6: DICOM Service Object Pair 15
Figure 3.7: Example SOP Instance 15
Figure 3.8: Storage Of One Image File 16
Figure 3.9: Impact In The Application Layer 17
Figure 3.10:The OSI Layers 18
Figure 3.11: Purpose of messaging standards 19
Figure 3.12: HL7 standard 20
Figure 3.13: PACS Components 27
Figure 3.14: Generic PACS Components and Data Flow 28
Figure 3.15: PACS Based Workflow 30
Figure 3.16: Framework Components For Portage 32
Figure 3.17: RAD-4 User Case Roles 33
Figure 3.18: IHE Technical Framework Development Process 35
Figure 3.19: 2021 Best in KLAS Ranking 36
Figure 4.1: Patient ID Mismatch Flowchart 39
Figure 4.2: Patient ID, Patient Name, Date of Birth Mismatch Flowchart 40
Figure 4.3: Workflow After First Mismatch 41
Figure 4.4: Block Diagram of DBMS Based Information System 42
Figure 4.5: Powershell Commandline 44
Figure 4.6: Interaction Between Two Entities Implementing HL7 V2.3 45
Figure 4.7: Interaction Diagram 47
Figure 4.8: Interaction Diagram 48
Figure 4.9: Interaction Diagram 49
Figure 4.10: Flowchart of the Mismatch Estimation Tool 50
Figure 4.11: Flowchart of the Main Processing 53

List of Tables

Table 3.1: Major functions of the PACS server and archive ……………………..…..……………..…29

Table 4.1: ORM message structure ……………………………………………………..………………...46

Table 4.2: MLW Keys for Query by Patient…………………….……………………….…………………..47

Table 4.3: MWL Keys for Query by Patient.……………..……………....48

Table 4.4: Example of output from the estimation tool...54

Abbreviations and Symbols

PACS	Picture Archiving and Communication System
EHR	Electronic Health Records
RIS	Radiology Information System
HIS	Hospital Information System
SQL	Structure Query Language
DICOM	Digital Imaging and Communications in Medicine
ACR	American College of Radiology
NEMA	National Manufacturers Association
VR	Value Representation
VPN	Virtual Private Network
SCU	Service Class User
SCP	Service Class Provider
HL7	Health Level 7
OSI	Open Systems Interconnection
RQ	Request Message
RSP	Response Message
DIMSE	DICOM Message Service Elements
SOP	Service Object Pair
UID	Unique Identifier
DBMS	Database Management System
IHE	Integrating the Health Enterprises
PID	Patient Identifier
AE	Application Entity
RDMA	Remote Direct Memory Access
ORM	Order Message
IoT	Internet of Things
M2M	Machine-to-Machine
IT	Information Technology

Chapter 1

Introduction

1.1 Motivation

The goal of this section is to get a better understanding about the subject of patient misidentifications and to highlight the current challenges of this area. Therefore, an illustration regarding: the challenges of patient matching, the resulting consequences of patient mis-identification as well as the most common root causes of these mismatches can be found below.

1.1.1 The challenge of patient matching:

With the latest advancements in technology and the resulting digitalisation of goods and services, it was only a matter of time for the healthcare industry to adapt in accordance with that.
Based on these new technological advancements, we nowadays have external imaging data which can be transferred into the local network of a healthcare enterprise. This data transfer is done via a medical image exchange platform that is usually deployed across a healthcare enterprise. Because of this exchange platform, the users of the various medical specialities can directly access and view the various studies of patients coming from different sites. Additionally, the content from physical media such as a CD or DVD can be added to the patient record and imported to the picture archive and communication system (PACS) (Jackie Leckas, 2021).

Due to this innovative way of sharing and accessing patient data, the clinical efficiency has drastically increased over the last decades. The digitalization of patient information has significantly reduced the duration of medical management as well as the diagnosis procedures. By sharing interoperable exam data across networks, and by having multiple healthcare enterprise users accessing external studies, it is crucial to have an exchange technology which verifies if the patient data is matching the referring exam study. As a result, the exchange technology must guarantee that clinical end users cannot link an outside study to a wrong record in the electronic health record EHR or PACS. Without such a precautious tool, the chances of having an incorrect association between a patient record and the linked exam studies could be relatively high. Hence, the various system end users would

be viewing data from the wrong patient (Adelman *et al., 2019*; Berkowitz *et al., 2018*; Huang, 2010; Imkamp *et al., 2016*).

As for now, previous studies have proven that medical error regarding therapeutic and diagnostic interventions can occur with a frequency between 0.005 and 0.08% (Just *et al., 2016*). In 2016, the Joint Commission (JC) listed a number of medical incidents reviewed by the Commission throughout the years 2005-2016. According to their findings, 12.7% of the medical incidents were based on identification errors (Erin Benson, 2017).

As the percentage of patient identification errors is still relatively high despite the current technological advancements, we can clearly assume that the management of patient matching needs to be improved.

1.1.2 Consequences of patient mismatches:

Accurate person identification has become essential throughout most human activities such as security checks, tax payment and especially health care. In general, the failure to identify patients correctly throughout the healthcare industry has regularly been associated with the risk of serious harm (Lippi et *al., 2017*).

Patient mismatches can lead to redundant tests, incorrect treatment services, unnecessary hospitalizations, wrong diagnosis as well as a significant increase of additional costs (Erin Benson, 2017). Worst-case scenarios of patient misidentification include: missed cancer, unnecessary mastectomy and hemolytic transfusion reactions. Misidentification of specimen, such as unlabelled or mislabelled specimens appear at a rate of 0.04% to 0.1% (Morrison *et al., 2010*).

Another example regarding the importance of accurate patient identification can be seen in anesthesiology. Anesthesiologists use drugs where the therapeutic indices are very narrow. In this case, a patient misidentification can lead to a misidentification of allergies, use of the incorrect dosing weight, usage of the wrong medication administration and entering incorrect orders to the patients. As a result, one of the preventable errors can cause patient morbidity and mortality. Consequently, the prevention of these types of errors shall be considered of a task with high priority (Taieb-Maimon *et al., 2018)*.

According to the Institute Patient Safety Organization (ECRI), the majority of patient mismatches is being handled in early stages. However, up to 9 % of the mismatches were leading to patient harm or even death (Just *et al., 2016*).

If a patient misidentification is detected, the referring medical staff member needs to correct the mistake. This action causes some additional costs as the missing information needs to be re-inserted into the system which can take some time. In the USA only, the costs of handling patient misidentification are on average $17.4 million per year. To efficiently treat their patients, the physicians need to access the patient's full medical history. This means that the information about the previous diagnoses, lab results, medical images, family background as well as medication history need to be correct so that the physicians can conclude with an accurate and correct diagnoses (Hillestad *et al., 2008*).

As a result, we can conclude that the consequences of patient misidentification shall be considered as highly important as they do not only create potential extra costs for the healthcare sector, but especially because of the related healthcare issues of the affected patients.

1.1.3. Cause of patient mismatches:

The present work investigation mainly focuses on Picture Archiving and Communication Systems (PACS), which are most commonly part of the Radiology departments.

In a modern Radiology department, the medical equipment receives their worklist from the local Radiology Information System (RIS). This information is automatically updated for all patient demographics and examination information for each patient and study.

However, it is common that in some cases the equipment does not receive the information from the various management systems due to licensing or local networking issues. In these cases, the data needs to be manually inserted. The manual input of patient data appears to be one of the most common root causes of patient data mismatch.

In most of the cases, incorrect patient identification happens as soon as the patient information enters the system of the referring healthcare enterprise. Usually, for these cases, an employee on-site inserts the patient and study data manually. Being a human operation, the employee can mistype the patient's name, birth date, patient identifier or request number which will cause data mismatches. If an employee cannot find a record of a patient in their system, the value might be left empty, however this will cause the equipment to automatically generate an internal identifier or even leave it empty.

In case a duplicate patient record is used, it can cause a spreading of the patient history and data over several records of the exact same patient. As a result, each record indicates an incomplete snapshot of the patient's health and the related procedures.

In some cases, a change regarding the marital status of the patients and even a change of the address may lead to some uncertainty. For example, in case of a divorce, not only the marital status of a patient will change, but also the patient's name may change. Due to the change of the name, the patient may be unfindable in the system an eventually be unreachable for receiving medical notifications.

Therefore, based on the mentioned causes for possible patient misidentification as well as the related consequences, it is highly recommended to manage a patient mismatch in early stages (Adelman *et al., 2019*).

1.2 Purpose and main goal of the task

Patient demographics mismatch is a mechanism that Sectra PACS (medical imaging archive software) implements, in order to identify and avoid potential risk of placing medical examinations in the wrong patient.

When medical images arrive to Sectra PACS, their patient data is compared against the known information about the patient demographics, like patient name, patient birth date and patient sex. If there is a mismatch, then the examination is placed in a separate list, and is not shown to the clinicians, until someone verifies the patient identity.

In cases where there are legacy PACS systems being migrated to Sectra PACS, this patient demographics mismatch could be raised thousands of times, but then the PACS responsible would have a number of mismatches to solve that is greater than their actual capacity.

In order to avoid this, the goal of the project would be to create a tool, that would compare the patient demographics in exams from legacy systems with a set of reference data, providing the required statistics about mismatches, before this information is actually sent to the Sectra PACS and real patient data mismatches are raised. This way, the project team (either Sectra's or the Hospital's) can take a decision on how to cope with them (Sectra, 2019).

1.3. Dissertation Structure

In addition to the introduction, this dissertation contains 4 more chapters.

Chapter 2 illustrates the schedule and the main tasks of the internship at Sectra.

In chapter 3, the background theory regarding patient data, the Radiology department and PACS are being illustrated. Additionally, the importance of a standardized health environment is explained. A brief introduction about Sectra and their product can be found in this chapter as well.

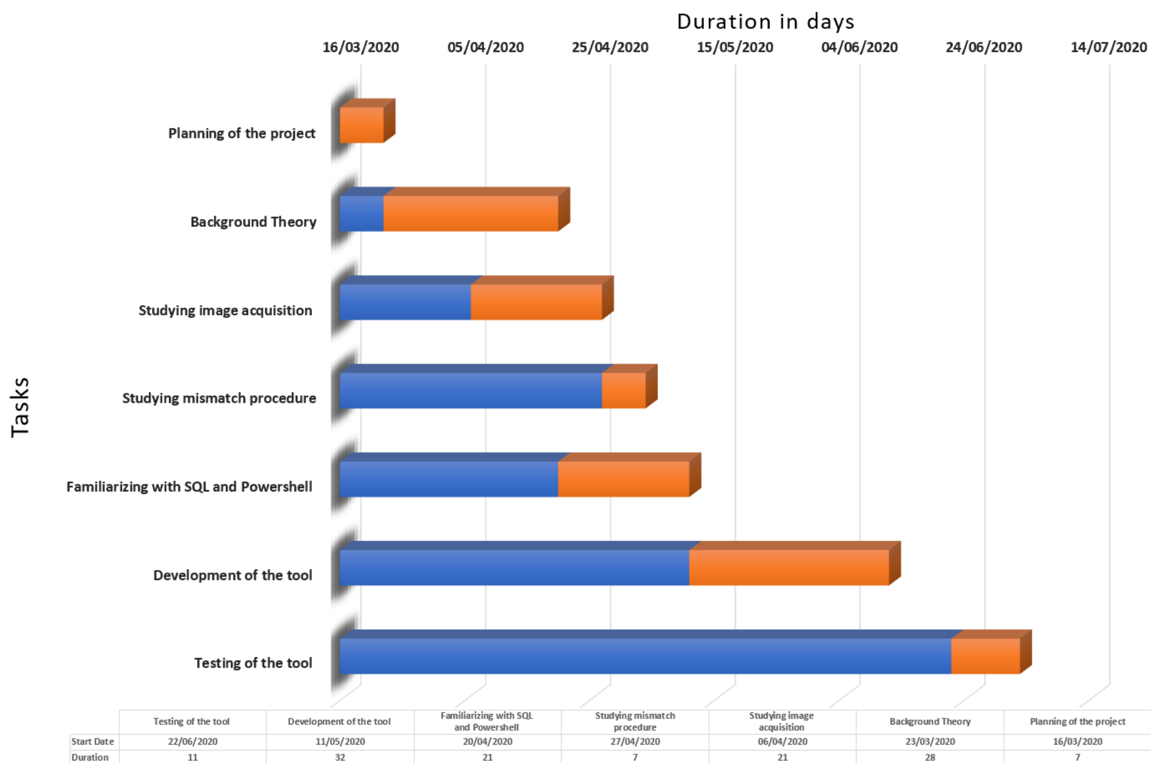
Chapter 4 contains all the information regarding the tasks and duties of the internship at Sectra. The fluxograms as well as the various procedures used for creating the examination mismatch tool can be found in this chapter.

Afterwards, the findings from chapter 4 are being discussed in chapter 5. This chapter contains the discussion part as well as the conclusion regarding the goals and achievements of the internship and the mismatch estimation tool. Furthermore, possible future work and issues are also discussed in this section.

The appendix contains all the Powershell scripts and SQL queries used to create the mismatch estimation tool. In addition, the fluxograms and diagrams created during the internship can be found there as well.

Structure and Schedule of the Internship

This section illustrates the structure of the internship at Sectra. Figure 2.1 represents a Gantt chart containing the main tasks of the internship and their duration.



	Testing of the tool	Development of the tool	Familiarizing with SQL and Powershell	Studying mismatch procedure	Studying image acquisition	Background Theory	Planning of the project
Start Date	22/06/2020	11/05/2020	20/04/2020	27/04/2020	06/04/2020	23/03/2020	16/03/2020
Duration	11	32	21	7	21	28	7

Figure 2.1: GANTT CHART – TASKS INTERNSHIP

As can be seen on figure 2.1, the internship was divided into seven main tasks. A description of each task can be found below.

Planning of the project:

The initial task was to elaborate an action plan regarding the development of the estimation tool. As the development of the tool required some knowledge regarding medical IT as well as some scripting, it was necessary to build a schedule which would guarantee that all the important topics were taken into consideration.

Background Theory:

This phase covers the theoretical part of the subject. It contains all definitions and illustrations needed to proceed with the development of the estimation tool. Therefore, the main task of this phase consisted of gathering all information and to document the findings in a most comprehensive way.
The subjects covered in this chapter are: DICOM, PACS, HL7 and IHE. The entire output of this task can be found in chapter 3.

Studying image acquisition:

As the examination mismatch occurs during image acquisition, a deep investigation regarding this subject was made. The main goal of this task was to create a flowchart which would represent a general workflow regarding image acquisition. The output of this research can be found in chapter 4 under the following subtitle: *4.2.3 Image acquisition workflow according to IHE standards.*

Studying mismatch procedure:

Since the purpose of the estimation tool was to detect examination mismatches before any image acquisition would take place, it was important to do a profound research regarding the examination mismatch process. As a result, the main task was to deliver a flowchart which would explain the exact mismatch mechanism. The result of this task can be found in chapter 4 under the following section: *4.1.2 DICOM image mismatch procedure*

Familiarizing with SQL and Powershell:

To successfully create the estimation tool, a basic knowledge regarding SQL and Windows Powershell was needed. In this context, several exercices regarding basic Powershell scripting and SQL queries were made in the IT-Lab environment.

Development and testing of the tool:

The estimation tool was created, based on the flowcharts and the scripting exercices from the previous tasks. The development was done by using the IT-Lab environment provided by Sectra and by using anonymized patient data.

The related flowchart as well as the used scripts can be found in the Appendix and under the following section of chapter 4: *4.2.4 Creating the mismatch estimation tool*.

Chapter 3

Background Theory

The role of the background theory is to provide a global overview regarding patient data, the radiology department and the migration of patient data between two different entities. This chapter provides all the information needed for having a clear understanding regarding the methodology of the patients mismatch estimation tool.

3.1 Digital Imaging and Communications in Medicine (DICOM)

Since patient data is commonly expressed as DICOM files, it is important to give a detailed explanation regarding the structure and the usage of these files.

3.1.1 Definition

DICOM stands for "Digital Imaging and Communications in Medicine" and is an official standard regarding the management and communication in the domain of medical information and related data (DICOM Standards Committee, 2020).
Currently, this standard is mostly used for transmitting and storing medical images. These medical images can furthermore be integrated in medical imaging devices like for example, workstations, servers, printers, scanners or network hardware. Nowadays, the majority of modern hospitals is processing their data by using the DICOM standard. Due to the global usage of this standard, smaller doctor's and dentist's offices have been starting to implement DICOM as part of their daily workflow. This standard was officially published in 1993 by American College of Radiology (ACR) and by the National Manufacturers Association (NEMA) which, until this day, holds the copyright to the published standard (DICOM Standards Committee, 2020).
This specific standard is also known as the NEMA standard "PS3" and as the ISO standard 12052:2017 "*Health informatics – Digital Imaging and communication in medicine (DICOM) including workflow and*

data management". The original mission of this standard was to enable users to retrieve images and associated information from digital imaging equipment in a normalized format using point to point connection. Additionally, another initial reason for using this standard was to hide any differences between the image equipment manufacturers (National Electrical Manufacturers Association, 2020).
By now, the parts of the DICOM standard are being documented under 22 titles. The current edition of this standard can be freely accessed under the following website:

https://www.dicomstandard.org/current/ (National Electrical Manufacturers Association, 2020)

3.1.2 DICOM Vocabulary

In order to get a simplified overview of the complex medical environment, the "DICOM information model" has been introduced. This model is based on objects which can be found in the real world. As the user can relate each DICOM variable to an attribute from the real world, the user can visualize the different DICOM elements in a "simplified" way. An example illustrating the relation between the real world objects and the DICOM elements can be found in figure 3.1.

3.1.3 DICOM Information Model

In this model, all patients, studies, medical devices, among others, are indicated by DICOM as objects with specific attributes. As previously mentioned, the objects and attributes are standardized according to the DICOM Information Object Definitions (IOD).
For example, a patient IOD can be described by a patient name, sex, weight, age, ID, smoking status etc.

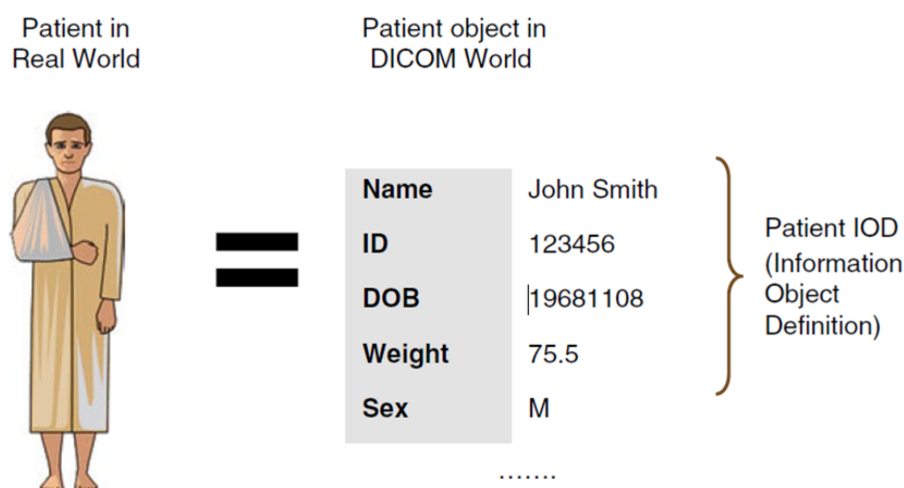


Figure 3.1: From real data to DICOM IODS (Pianykh, 2012)

To ensure a certain consistency regarding the process, the naming or the format of these attributes, the **DICOM Data Dictionary** has been created. This dictionary contains a list of more than 2000 standard attributes. In relation to figure 1, the patient's attributes such as: sex, name and date of birth are included in the DICOM Data Dictionary.

Overall, the DICOM attribute or data element is characterized based on four factors:

1. A tag which is indicated in the format of group, element (xxxx,xxxx). This tag identifies the element.
2. A Value Representation (VR). The VR describes the data type and format of the attribute's value.
3. A value field. This field contains the attribute's data.
4. A value length that defines the length of the attribute's value.

The basic attribute structure is shown in the following figure.

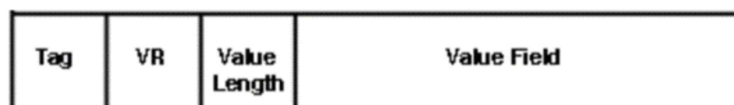


Figure 3.2: DICOM Attribure (Data Element) Structure (Darthmouth College, 2020)

Figure 3.2 illustrates the structure of a regular DICOM attribute. As indicated above, the order of a DICOM attribute, from left to the right is the following: Tag, VR, Value Length and Value Field.

3.1.4 DICOM Connections

Each DICOM data attribute is formatted based on 27 value representation (VR) types. These VR types contain dates, time, names etc.

The captured DICOM data attributes can furthermore be processed and even be sent within various application entities (AEs). These AEs are whether devices or software. Figure 3.3 indicates the most common examples of AEs.

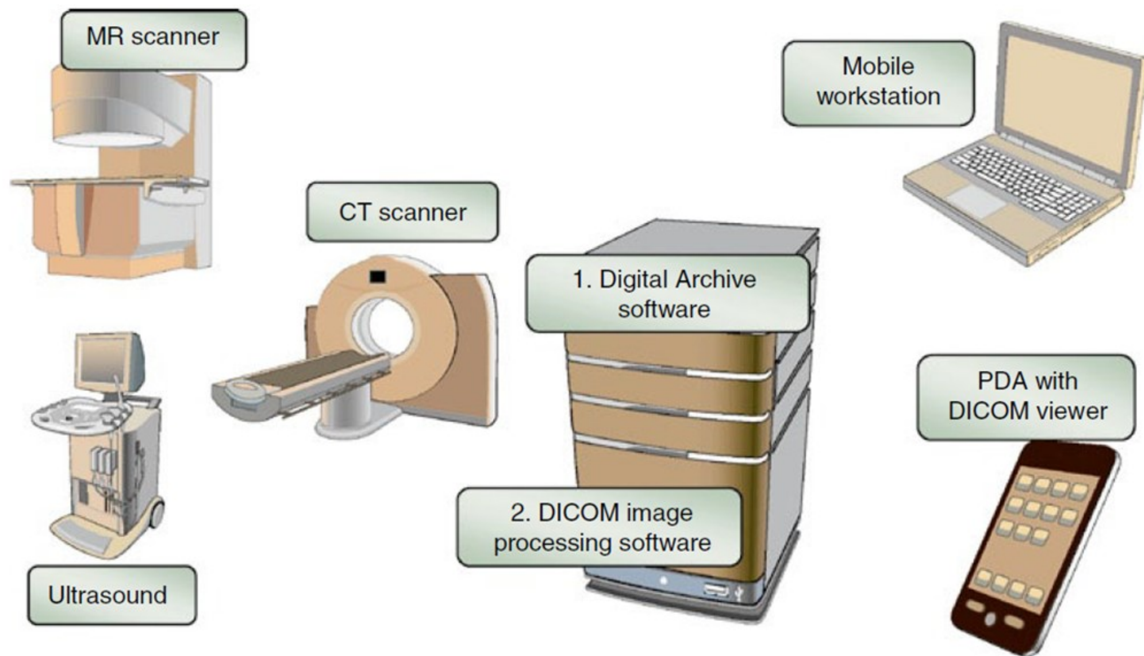


Figure 3.3: DICOM Application Entities (Pianykh, 2012)

As indicated on figure 3.3, AEs can be various modalities, workstations but also programs. This means that a single device can run several AEs.
For transmitting data, DICOM uses the Transmission Control Protocol (TCP) and the Internet Protocol (IP). The TCP/IP protocol is the same which is being used for sending emails or watching online videos. It accommodates all software and hardware variations and transmits the most fundamental network functionality. In this case the information is being sent as a sequence of bytes from one port/IP address to another. DICOM only adds its own networking language on the application layer of the OSI model.
The DICOM language consists of “high-level” services, also known as “DICOM DIMSE”. These services are built on “low-level” DICOM association primitives (DICOM upper layer protocol) (Pianykh, 2012).

3.1.5 DIMSE Services

To successfully transfer DICOM files, DICOM Message Service Elements (DIMSE) are being used. This means that DICOM AE’s use DIMSE for providing or requesting service information. Consequently, the protocol of DIMSE forms the basis for any service exchange regarding DICOM files.

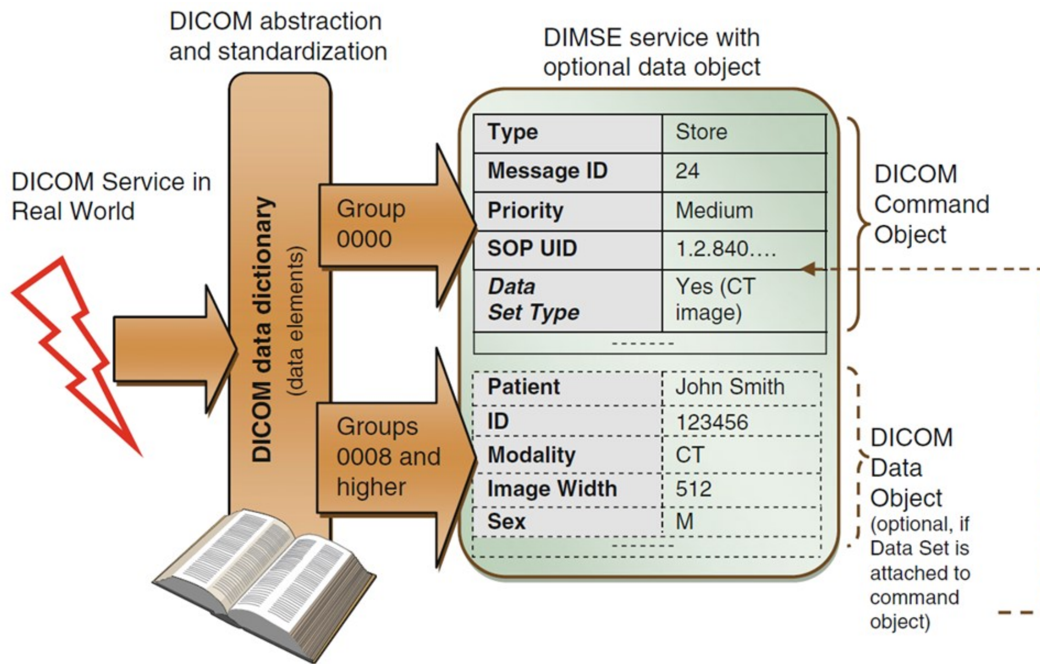


Figure 3.4: Structure of DICOM DIMSE (Pianykh, 2012)

All the DIMSE services have a response as well as a request/query component.
Hence, the service class user (SCU) AE's apply the requests whereas the responses are performed by the service class provider (SCP) AEs.
As can be seen in figure 3.4, the DICOM network objects are expressed in the same VR rules which were mentioned previously. However, there is a clear difference between DICOM data attributes and the service attributes. In fact, DICOM reserves a single "0000" group for all service tags.

Figure 3.5: Example On How DICOM Applications (AES) Communicate (Pianykh, 2012)

Figure 3.5 illustrates graphically how the interaction between a service class user (SCU) and a service class provider (SCP) is being performed. DICOM AEs send service messages to each other, containing requests (RQ) or providing operations. These requests/provided operations are the also called "DICOM Message Service Elements (DIMSE).

The most common DIMSE services (Pianykh, 2012):

- **C-Echo :**

  Is one of the most common used and also one of the simplest DIMSE service. Despite the basic function of this service, its use is essential in DICOM networks domain.
  Even though DICOM devices appear to be connected physically through a network cable, a physical connection is not sufficient. Additionally, the use of the command ‘ping’ (which indicates that the devices are TCP/IP – networked) is not a guarantee that the referred devices are DICOM-configured.
  As a result, the use of a ‘DICOM-Echo’ is highly recommended in order to verify if the various devices are able to communicate with each other.

- **C-Store:**

  This DIMSE is used to request the storage of a composed service object pair (SOP) instance information by a peer DIMSE service-user.

- **C-Find:**

  This command is used to match a series of string attributes against the attributes of the set of SOP Instances managed by a peer DIMSE service-user. The C-FIND service returns for each match a list of requested attributes and their values.

- **C-Move:**

  Used to move the information for one or more Composite SOP Instances from a peer DIMSE-service-user, to a third party DIMSE service-user, based upon the Attributes supplied by the invoking DIMSE-service user.

- **C-Get:**

  Used to fetch the information for one or more Composite SOP Instances from a peer DIMSE-service-user, based upon the attributes supplied by the invoking DIMSE service-user.

3.1.6 DICOM Service Object Pair

A Service-Object Pair (SOP) is defined as the combination of a DICOM Service Element (DIMSE) and an Information Object Definition (IOD)(see figure 3.6). The SOP Class definition contains the rules and semantics which may restrict the use of the services in the DIMSE Service Group or the Attributes of the IOD (Softneta, 2020).

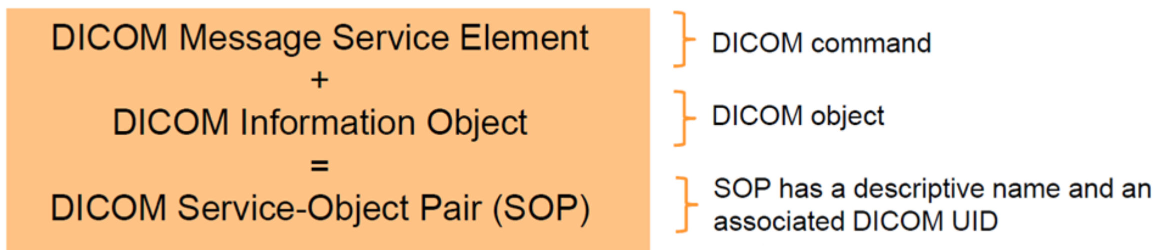


Figure 3.6: DICOM SERVICE- Object Pair (Pianykh, 2012)

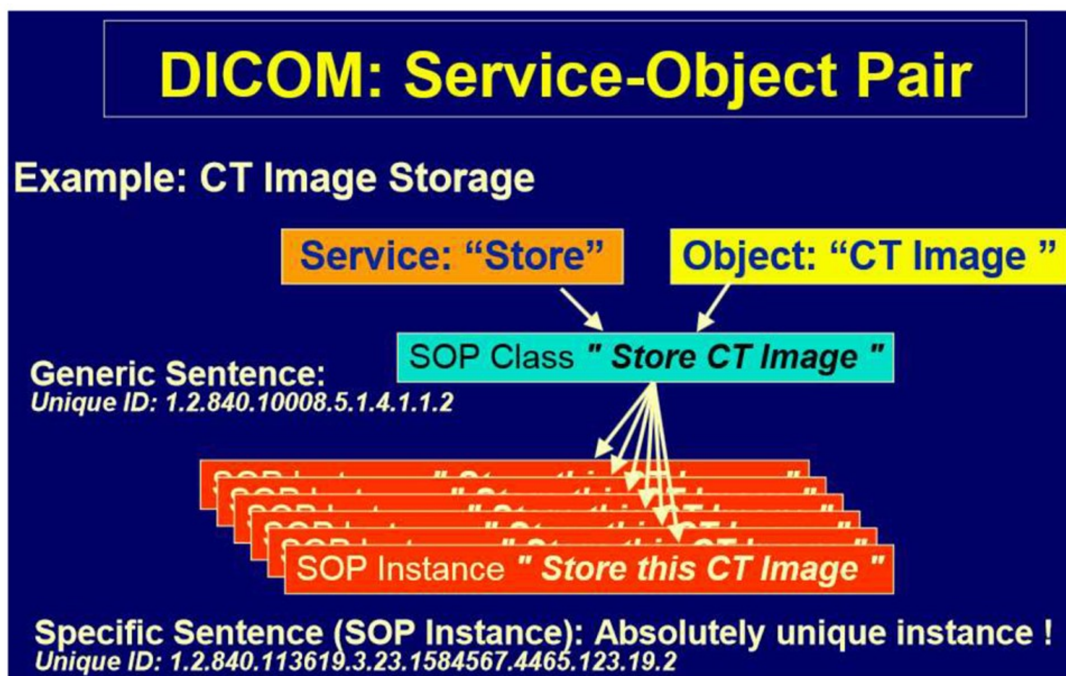


Figure 3.7: Example SOP Instance (Pianykh, 2012)

Figure 3.7 demonstrates how SOP instances are created. As previously mentioned, SOP classes are created by combining a DICOM Service with a DICOM Object. SOP Classes are furthermore subdivided into "SOP Instances". Each SOP instance DICOM file is absolutely unique. Figure 3.7 indicates an example whereas the SOP Class "Store CT Image" (SOP ID: 1.2.840.10008.5.1.4.1.1.2) is subdivided into various SOP Instances. In this case a SOP Instance stands for storing one specific image.

3.1.7 Putting all together

In order to clarify the usage of DICOM files and the resulting advantages of DICOM files, the following example will be introduced.

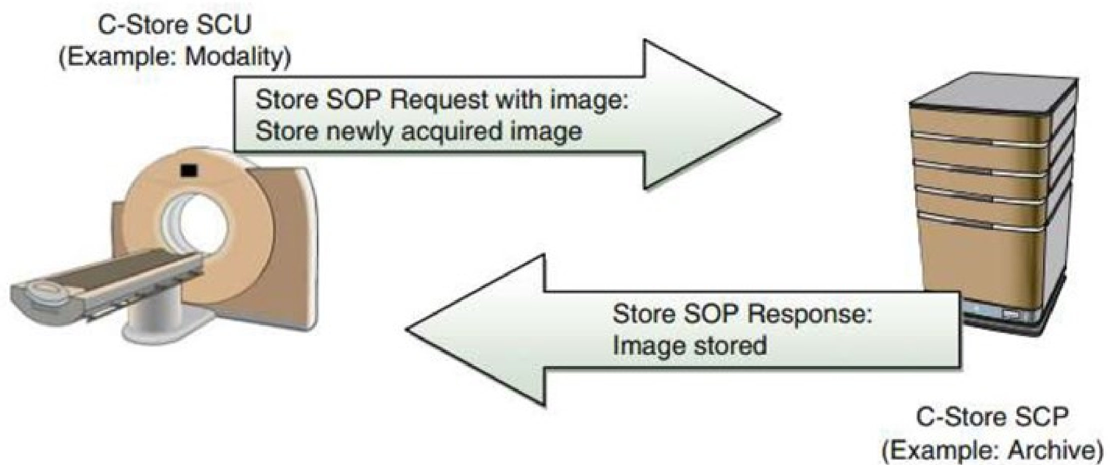


Figure 3.8: Storage of one image file (Pianykh, 2012)

Figure 3.8 represents the process regarding the storage of an image file. In this example the image file was created by an MRI modality. To successfully store the image in a database, the modality needs to apply a C-store command. As the modality is applying a C-store request towards an archive (the database), the modality is acting as a service class user (SCU). As the role of the archive is to actively store the image, the archive appears to be the service class provider (SCP). The arrows between the SCU and the SCP describe the request as well as the response of the service object pairs (SOPs).
After the transfer of such an image is done, the referring DICOM tags of the image is changed as well. The change of a DICOM tag is illustrated under the following figure indicated below.

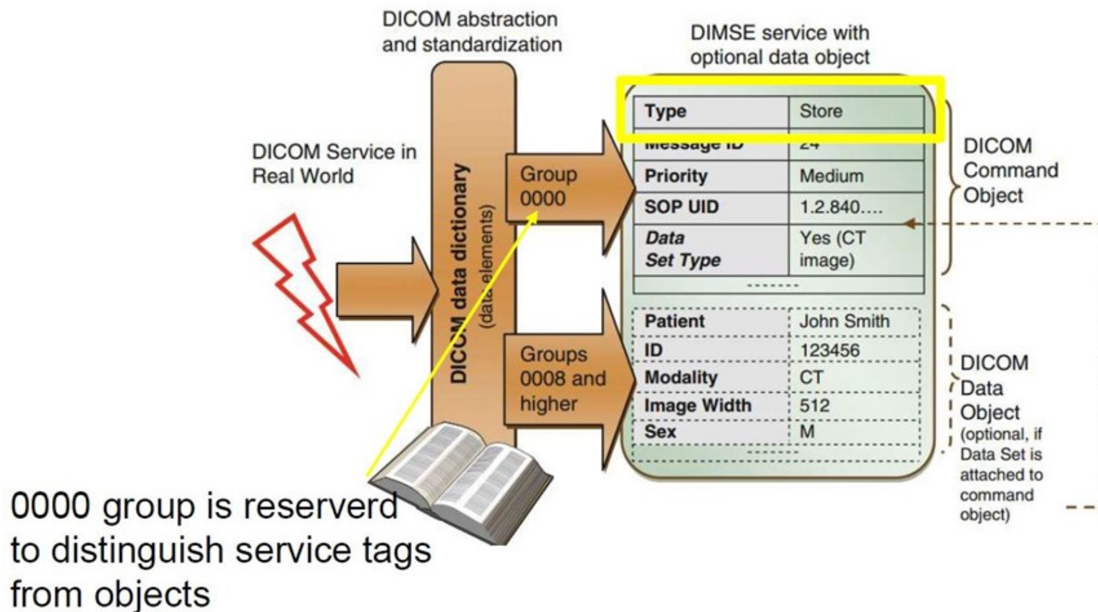


Figure 3.9: Impact in the Application Layer (Pianykh, 2012)

Figure 3.9 provides information about a DICOM tag which has been changed due to an image storage operation. After successfully storing the image in the archive, the image can not only be found in the referring database, but the information content of the stored file has been changed as well. As highlighted in figure 9, the type of the DICOM file has been set to “store”.

In relation to the internship project at Sectra, the usage of DICOM files as well as the various DIMSEs and the manipulation of those files through different types of command lines, were part of the main task. Therefore, to successfully proceed with the main project, a proficient knowledge of the DICOM standard was required.

3.2 The Health Level 7 Standard

Over the last decades more applications have appeared throughout the various departments of hospital environments. Therefore, it is important to maintain the data quality in this field as high as it currently is. To guarantee a transparent communication between the various entities, the communication should be made by the usage of a communication standard.

The most widely used standard for data exchange between healthcare applications is Health Level 7 (HL7) ("Errata," 2002; National Commitee On Vital And Health Statistiacs, 2000).

3.2.1 Definition:

Health Level Seven (HL7) is defined as a set of international rules, definitions and standards which are applied to transfer and to exchange medical data within health care providers. In general, this transfer is done via electronic health records (EHRs). HL7 has become the most widely used and therefore the leading syntactic standard in medical IT (Benson, 2012; Health Level Seven International, 2020) .

The exchange of HL7 data is executed on the seventh layer of the Open Systems Interconnection (OSI) model. As a result, the name of this syntactic standard (level 7) refers to the application layer of the OSI (Cloudflare, 2020).

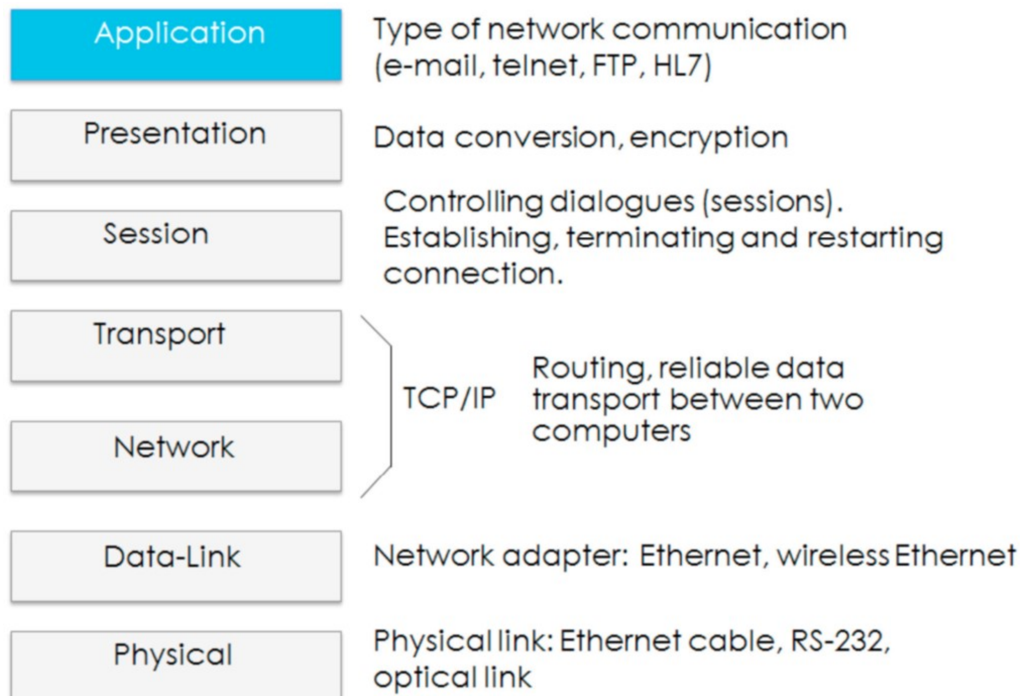


Figure 3.10:The OSI Layers (Shafarman et al., 2000)

Figure 3.10 represents the OSI model. As can be seen, there are 7 different layers.

The three first layers on the top are involved with the process of applications (interworking). The four lower layers are used for transmitting the data (interconnecting).
The layers represent the various levels regarding the transfer of data between entities. The application layer represents the direct interaction between the users and the referring software application. As already mentioned, HL7 is being used on the application layer (Benson, 2012).

3.2.2 Advantages of using a messaging standard:

To allow different computer systems to share patient information, the information needs to be transferred from one system to another. Before the interoperability standards such as HL7 were created, this transfer was generally done through customer interfaces (see figure 3.11) (Oliveira *et al.*, *2016*).

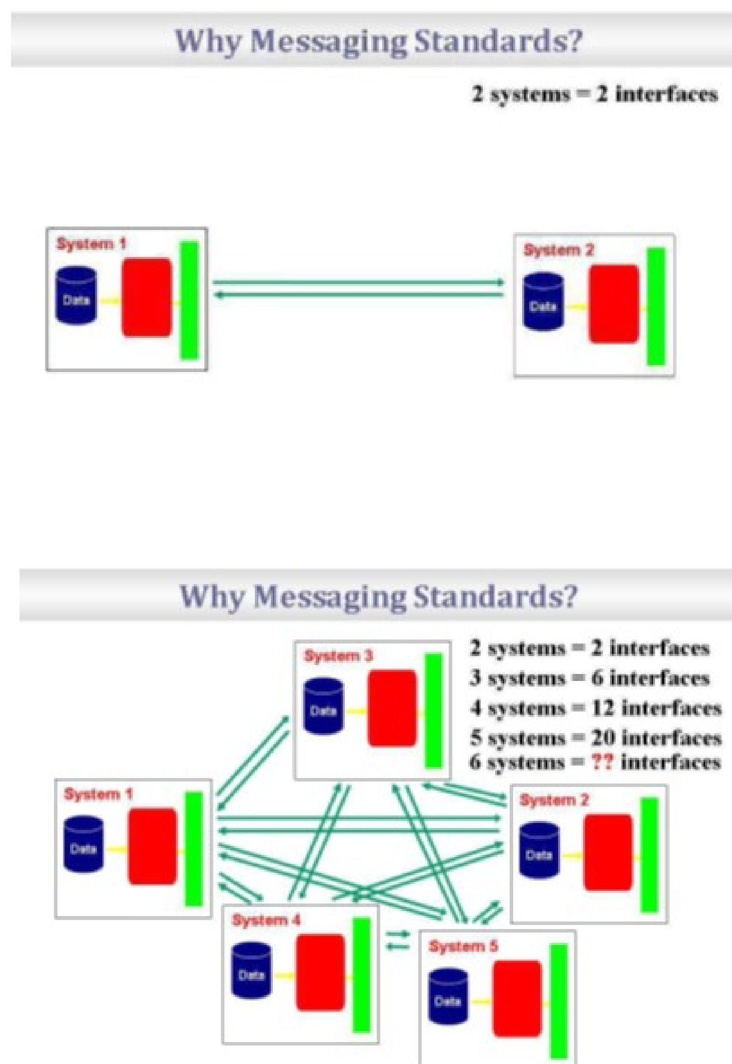


Figure 3.11: Purpose of messaging standards (Sectra, 2019)

Figure 3.11 illustrates the exponential increase of interfaces which are required for establishing a communication path between the various systems. As demonstrated on the scheme, the addition of

more systems is related to a bigger addition of the required interfaces. This increase can lead to a complex networking system which increases the risk of a less efficient communication between each system. Therefore, to minimize the amount of interfaces, the usage of a “hub” interface with a standardized syntax such a HL7 would be a possible solution (see figure 3.12) (“Errata,” 2002).

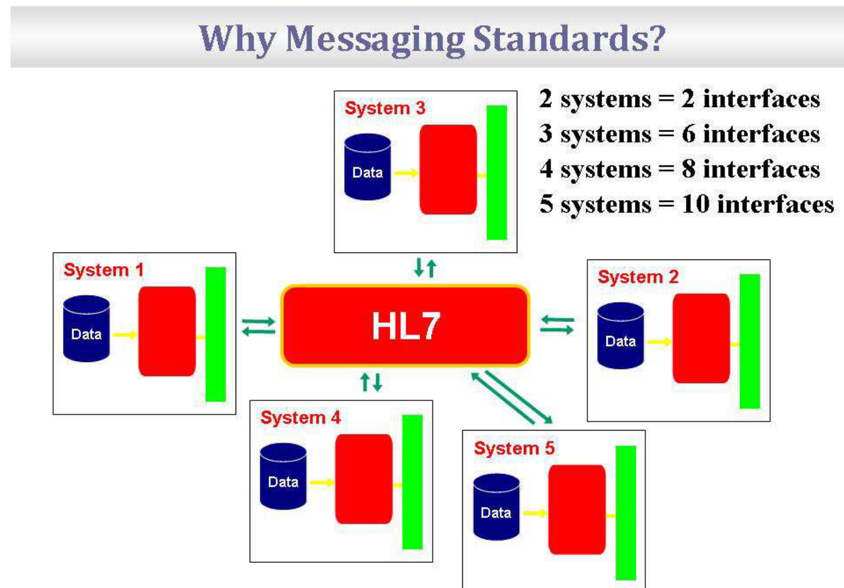


Figure 3.12: HL7 standard (Sectra, 2019)

As indicated on figure 3.12, the usage of the HL7 messaging standard reduces the amount of the resulting interfaces. This leads to a less complex system and therefore to a more simplified overview regarding the exchange of patient data.

3.2.3 Versions of the HL7 Standard:

Currently, the HL7 Standard is split into two versions, the version 2 and version 3. Most of the HL7 messages use the versions 2.3 and 2.3.2 of the standard. The version 3 of the HL7 standard is only used in a small portion for medical devices.
Simultaneously HL7 is working on a different messaging standard, called “HL7 FHIR” (Benson, 2012; Huang, 2004; National Commitee On Vital And Health Statistiacs, 2000)

HL7 V2.x:

The V2 standard contains 80 percent of the interface framework. The other 20 percent can be customized by the local implementation.
Overall, the 2.x versions are backwards – compatible with previous versions. The reason for this is that the V2 standard can disable unexpected message elements. As a result, an older device is able to process messages from devices which are running a newer application using a more recent HL7 version (Kiourtis *et al., 2019*).

HL7 V3:
This version was released in 2005. Even though the government and medical information users have promoted and influenced this version, it was less popular among the clinical interface specialists. The HL7 V3 is not backwards – compatible with V2 versions of the standard. This means that existing V2 interfaces cannot communicate with interfaces using the HL7 V2 version, unless considerable modifications are made.
The main goal of the Hl7 V3 was to globally increase the adaption of the standard by defining a consistent data model and by creating an overall more precise definition of the standard *(*Krishna *et al.,2019)*

FHIR:
Fast Health Interoperable Resources (FHIR) is the latest standard framework created by HL7. This standard combines the advantages of the previous HL7 versions and uses the latest web service technologies. The significant difference between FHIR and the previous HL7 versions is the fact that it is based on RESTful web services as well as on modular components known as “resources”. This messaging standard supports JSON and XML making it compatible for the majority of the current applications (*Proceedings of the 26th IEEE International Symposium on Computer-Based Medical Systems*, 2013 - 2013).

3.2.4 Application of HL7:

Overall, the basic data unit in HL7 appears to be a message. These messages contain various segments which are assembled in a specific sequence. A segment consists of several data fields whereas each segment is expressed by its unique and predetermined code of three characters. An example of the most frequently used three characters code can be seen below:

- MSH : Message Header Segment
- EVN: Event type Segment
- PID: Patient identification Segment
- NK1: Next of kin Segment
- PV1: Patient visit Segment

As indicated above, the “MSH” three letter code stands for the message header segment. This segment indicates the destination, the source, the intent, the time stamp and even further information. The symbols for declaring the separation between each field and even within a field are also defined in the message header. The “PID” contains demographic information about the patient such as name, id code, address and so on. “PV1” contains information regarding the patient’s stay in the hospital such as location assigned, referring doctor etc. The following example represents an HL7 transaction of admitting a patient for surgery:

(1) Message header segment
MSH||STORE|HOLLYWOOD|MIME|VERMONT|200305181007|security|
ADT|MSG00201|||<CR>
(2) Event type segment
EVN|01|200305181005||<CR>
(3) Patient identification segment
PID|||PATID1234567||Doe∧John∧B∧II||19470701|M||C|3976 Sunset Blvd∧Los Angeles ∧CA∧90027||323-681-2888||||||||<CR>
(4) Next of kin segment
NK1|Doe∧Linda∧E||wife|<CR>
(5) Patient visit segment
PV1|1|I|100∧345∧01||||00135∧SMITH∧WILLIAM∧K|||SUR|ADM|<CR>

By combining all the five segments together, the resulting HL7 message expresses the following statement:

“*Patient John B. Doe, II, male, Caucasian, born on July 1, 1947, lives in Los Angeles, was admitted on May18, 2003 at 10:05 a.m. by Doctor William K. Smith (#00135) for surgery. The patient has been assigned to Room 345, bed 01 on nursing unit 100.The next of kin is Linda E. Doe, wife. The ADT (admission, discharge, and transfer) message 201 was sent from system STORE at the Hollywood site to system MIME at the Vermont site on the same date two minutes after the admit.”*

The “|” is the data file separator. If no data are entered in a field, a blank will be used, followed by another “|.”
The data communication between a hospital information system (HIS) and a radiological information system (RIS) is event driven. This means, that when a patient admission, discharge or transfer (ADT) event occurs, the HIS would automatically send a broadcast message, via an HL7 message, to the RIS. The RIS would then parse this message and insert, update, and organize the patient demographic data in its database according to the event. Similarly, the RIS would send an HL7-formatted ADT message, the examination reports, and the procedural descriptions to the PACS. When the PACS had

acknowledged and verified the data, it would update the appropriate databases and initiate any required follow-up actions (Huang, 2004).

In relation to the tasks at Sectra, a basic knowledge of the HL7 messaging standard was needed to fully understand the PACS workflow. As the communication between the various entities is partially done via HL7, a PACS user needs to understand the role and the importance of this messaging standard so that he/she can solve any potential issues.

3.3 Radiology Department

In 1895 the X-Ray was discovered by accident under an experiment of Wilhelm Röntgen. These X-rays were the result of a discharging electrical current in a highly-evacuated glass tube (Martins, 2014). As this discovery made it possible to minimize the “diagnostical subjectivity” from the clinical signs and as therefore a “new way to look into the body” was created, we can claim that this discovery had a revolutionary impact in medicine (Mazzoncini de Azevedo-Marques & Covas Salomão, 2009).
During the 1970’s, the ultra-sonography (US) was introduced, a new innovation which was exploring the ultra-sounds properties. As a result, a new way for obtaining medical images was created. This new technique had a tremendous impact on how medical images were created. Especially, as real time examinations were visible to the US-users (Dicken *et al., 2017*).
The main tasks of the Radiology department are the diagnosis and the report of the findings to the clinicians. Nowadays, it is the radiologist who is in charge of the imaging reports. This means that the radiologist analyses the medical images and elaborates a conclusion based on the findings (e.g., radiology report). The radiology report contains relevant information which is then sent to the referring clinician. Based on the given information, the referring clinician takes the necessary patient care decision (e.g., surgery).

3.4 Hospital Information System (HIS)

The HIS was introduced in the 1960’s. The role of the HIS relies on the management of patient data, invoicing and the related patient visits in the HealthCare institution (*Practical Guide to Clinical Computing Systems*, 2015).
With the creation of this system, the following applications were made possible:

- The creation of a unique patient identifier used in all the institution applications (Master Patient Index)
- Admission of patients and management of their administrative data.
- Reception of the payments and management of the related cash flow.
- Scheduling of the laboratory/radiology examinations.

For now, the HIS includes all sorts of administration regarding the financial and clinical application of a medical institution. Thanks to this system, the related patient data can be managed within multiple departments such as: the clinical, nursing, radiology, finance and laboratory department. One of the initial roles of the HIS was the integration of the patient data within all the hospital applications. Therefore, the collection of all demographic, financial and clinical information of each patient would offer the following advantages (Dixon, 2016; *Practical Guide to Clinical Computing Systems*, 2015):

- A reduction of transcription errors
- A reduction of an ambiguous information entry
- An enhancement of the information integrity

3.5 Radiology Information System (RIS)

The RIS is part of the radiology management. The main goal of this information system is the support of the operational workflow. The RIS can be seen as a repository of patient reports and patient data. Additionally, this system is not only used in medical imaging but also in different medical departments such as nuclear medicine and radiotherapy.
Overall, the RIS is normally associated with the following functionalities (Dixon, 2016; *Practical Guide to Clinical Computing Systems*, 2015).

- Patient consent and user authentication

  - In general a user is allocated to a role. These roles have specific permissions such as signing reports, view clinical information etc.
  - After an authorization regarding certain medical acts is made, the patient consents are stored.

- Demographics and alerts of patients

  - The patient demographics shall always be updated and have a Master Patient Index (MPI) or a Patient Administration System (PAS) as on of the main references.
  - Critical patient information such as infections or allergies must be exchanged and stored together with additional care record systems.

- Paper – and electronic based requesting

  - The RIS should be able to process the referrals coming from different electronic remote requesting systems (ERR)
  - The RIS should be able to process manual referrals

- Scheduling and Booking

  - Imaging and non-imaging bookings and schedule procedure steps (e.g. a radioisotope injection and a radioisotope image) shall be managed by the RIS.

- Folders and Worklists

  - A Worklist contains a list of examination procedures required for defined work cases.
  - A folder is a directory used to group examinations.

- Examination details

    - The RIS should contain all necessary information related to the given procedures such as dosage, technique, persons involved in the procedure as well as their role etc.

- Reporting

    - All aspects regarding the reporting workflow have to be supported by a RIS. This means reporting through dictation, word processing and report verification as well as issuing shall be supported.

- Image tracking
- Stock control
- Billing
- Integration

    - The transactions between systems must be set according to the suggestions of the IHE. This means that functional integrated solutions shall be achieved.

3.6 Picture Archiving and Communication System Components and Workflow

The picture archiving and communication systems (PACS) is a technology which is being used in today's medical imaging. The main role of the PACS is to provide an economical storage as well as to access the stored medical documents coming from various modalities, in a more convenient way (Choplin *et al., 1992*).

The medical images which are formatted in an electronic way and the reports of patients are processed digitally through the PACS. As a result, the process of manually retrieving and sending/transporting film jackets have been eliminated. The elimination of this manually and more time consuming process has not only made the usage of the PACS more popular, but it has revolutionized the efficiency regarding the process of medical imaging.

The format of the files which are being processed via PACS is the DICOM format.

3.6.1 PACS Components

In general, a PACS is composed of a data and image acquisition gateway. Apart of that, a typical PACS is supposed to contain a PACS server as well as an archive. To visualize the content of the various file, several display workstations are integrated together via an integrated digital network. Additionally, it is common that a PACS is connected to a healthcare information system (HIS) or a radiological information system (RIS) via database gateways and communication networks.

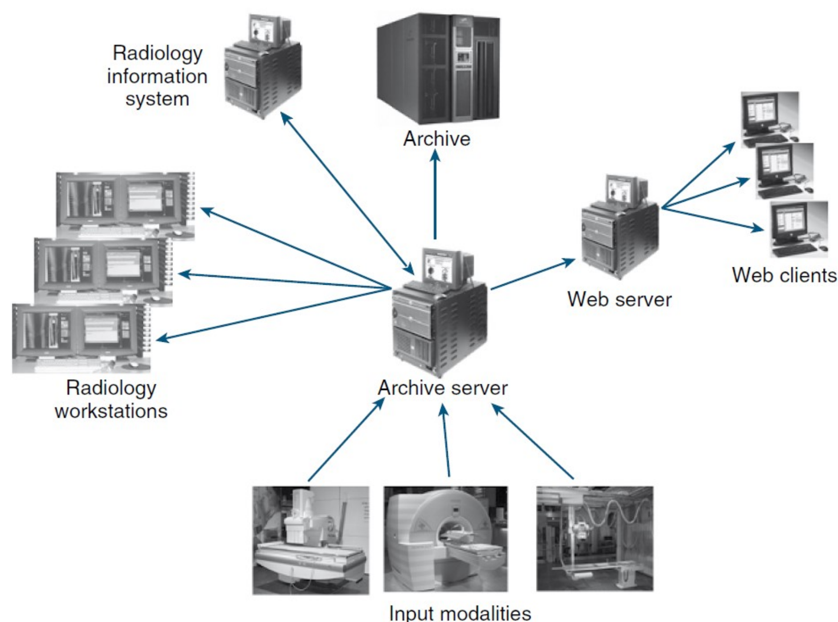


Figure 3.13: PACS Components (Branstetter, 2010)

Figure 3.13 represents an overview of the most common PACS components. As illustrated, the "archive server" appears to be in the centre of the entire system. This means that all the information which us being processed throughout the entire system needs to pass through the archive server.

The images acquisition is one of the main tasks in a PACS. As the imaging modalities can be supplied by different manufacturers, it is important to check the DICOM compliant statements of the modalities as well as for the PACS. If the DICOM compliant statement between the PACS and the modalities is not synchronized, a configuration cannot take place.

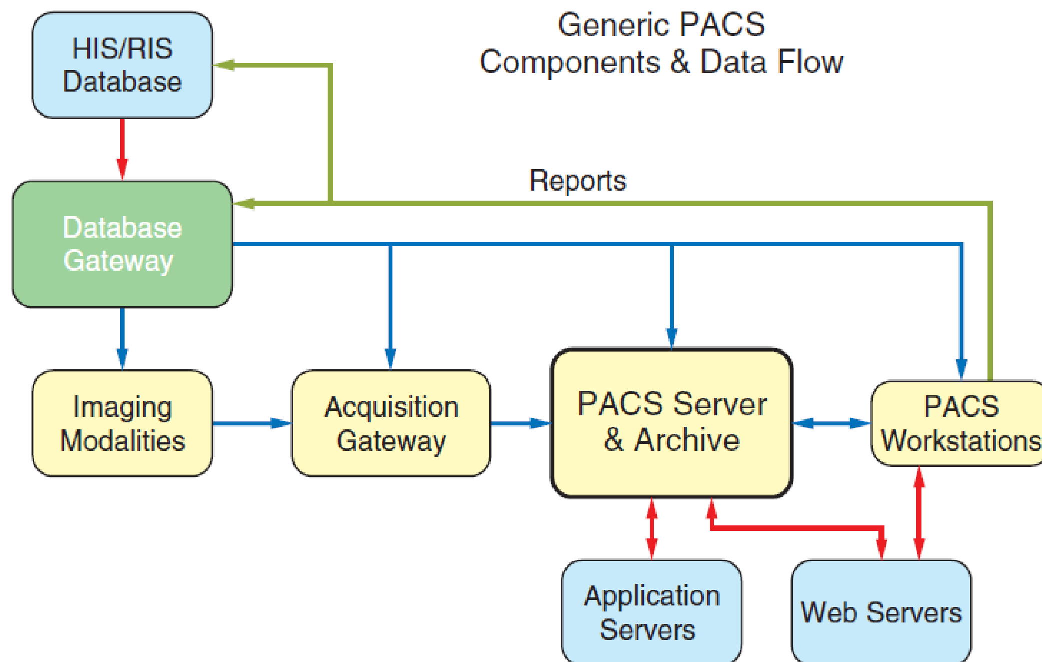


Figure 3.14: Generic PACS Components and Data Flow (Huang, 2010)

Figure 3.14 represents the regular PACS components in yellow. The internal data flow between the PACS components is indicated in blue, whereas the dataflow between the PACS and other information systems is indicated in red and green. As indicated on the figure and as mentioned previously, the PACS receives the initial data via the database gateway.

3.6.2 Functionality of PACS

As indicated on figure 3.14, the patient data as well as the related examination information are transmitted to the PACS. This information is originally coming from the HIS and the RIS where it firstly passes the database gateway before being sent to the PACS. This step is highly recommended in order to fully synchronise the various modalities, the HIS, the RIS and the PACS with each other.
The PACS server can contain various high-end computers or servers. The PACS server appears to be the engine of the system. The two components of the PACS archive and server are: a database server and a related archive system. The tasks for the archive system are to host the files for a short-term, long-term and even a permanent duration.
The main functions of the PACS server and archive are summarized under the following table:

Table 3.1: Major functions of the PACS server and Archive (Huang, 2010)

Major functions of the PACS server and archive

- Receives images from examinations (exams) via acquisition gateways
- Extracts text information describing the received exam from the DICOM image header
- Updates the database management system
- Determines the destination workstations to which newly generated exams are to be forwarded
- Automatically retrieves necessary comparison images from historical exams from a cache storage or long term library archive system
- Automatically corrects the orientation of computed or digital radiography images
- Determines optimal contrast and brightness parameters for image display
- Performs image data compression if necessary
- Performs data integrity check if necessary
- Archives new exams onto long-term archive library
- Deletes images that have been archived from the acquisition gateway
- Services query/retrieve requests from WSs and other PACS controllers in the enterprise PACS
- Interfaces with PACS application servers

3.6.3 A Generic PACS Workflow

According to the Cambridge Dictionary, workflow is defined as the way that a particular type of work is organized, or the order of the stages in a particular work process (Cambridge Dictionary, 2021). In radiology the term "workflow" is used to describe the order in which the entire examination process is applied.

In terms of radiology, workflow illustrates the different stage of an examination. These stages start at the order entry of the examination and mostly finishes at the transcribed report.

Even though the generic radiology workflow should keep the same order of the main stages, it is important to mention that each radiology department has its own workflow. As there are several parameter which can influence the workflow, such as the various types of modalities, the distance between the stations on site etc, it is necessary to observe the various radiology sites before elaborating a workflow procedure (Branstetter, 2010).

Regarding the PACS workflow, the technologist normally receives the order of a planned examination via an electronic worklist (see figure 3.15).

- The order is usually inserted in the RIS. The RIS transfers the message to the PACS where all the historic images are retrieved and put on the short-term archive. On the contrary to the film-based workflow, there is not waiting for the file room to retrieve a film jacket from the off-site storage location.

- Then, the technologist prepares the room and retrieves the patient where the patient history is being performed. In general, the history is inserted electronically into the computerized medical record of the patient.
- After the insertion of the patient history, the technologist performs the examination. The patient images are then sent to the appropriate PACS destination and are tagged with information from the RIS. Due to these tags, the historic image reports are available in the PACS in case new images are sent.
- The images are either pulled by the radiologist via an electronic worklist or taken to the radiologist. Usually, the radiologist views historic images and compares the previous images with the current images.
- While observing the images, the radiologist can dictate a report. In this case, a voice recognition software can be used, if a voice recognition software is used, the user can review the report right after the dictation and apply some possible corrections.

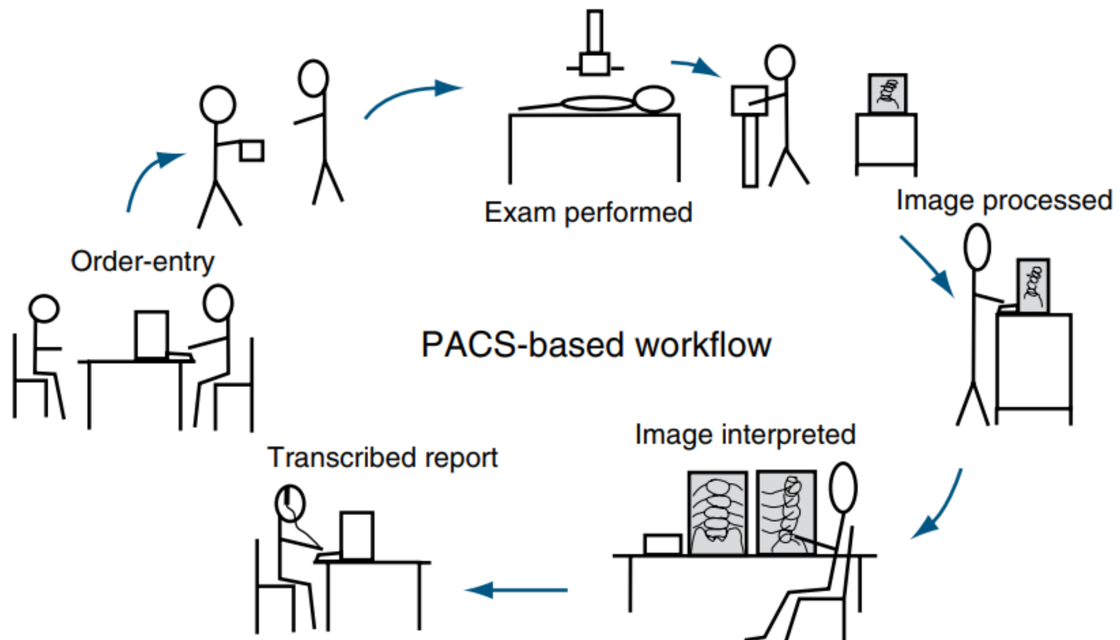


Figure 3.15: PACS Based Workflow (Branstetter, 2010)

Since the usage of PACS, the duration between the performance of the examination and the completion of the final report has been reduced significantly. In general, the completion of a PACS based workflow lasts only a couple of hours whereas the completion of a film-based workflow can take up to several days.

3.7 Migration of Medical Images and Related Data

3.7.1 Definition

Data migration is defined as the procedure of selecting, transferring and transforming data from a legacy data source to a new data source. Furthermore, the verification for completeness of the transferred data as well as the deactivation of the legacy storage system are considered as a part of the data migration process (Morris, 2012; Redbooks, 2015).

3.7.2 Purpose of Data Migration

There are multiple reasons for applying data migration. Overall, the main reasons for migrating data to a different source are the following:

- Maintenance of the legacy server
- Relocation of the data centre
- Consolidation of the related website
- Server or storage equipment replacements or upgrades

In order to minimize the impact on business operations during a migration project, it is important to plan the process in a most effective way. This planning includes the consolidation of the technology, its implementation as well as the validation of the technology. Therefore, a deep understanding of the design requirements is essential. As a result of the planning process, the organization in charge of the migration generally communicates the methodologies behind the project. This includes the installation of the software for the migration and the configuration of the necessary hardware (Techopedia, 2017; Thalheim & Wang, 2013)

The more the data migration is automated, the less human intervention as well as less application downtime is needed. Hence, the migration speed increases. The documentation of a migration project allows the parties to track potential incidents which may reduce future migration risks and costs (Kim, 2005; Schreiber & Garber, 2020).

As soon as the migration project is finished, the data should be validated for determining the level of accuracy. The last step of a typical migration project consists of data cleaning. This means that redundant data shall be removed. This action should be taken so that data quality can be improved by deleting repetitive or unnecessary data (Berkowitz *et al., 2018*; Teli *et al., 2016*).

3.7.3 Data Migration Technique

As indicated on figure 3.16, a data migration and transformation typically involves 4 steps:

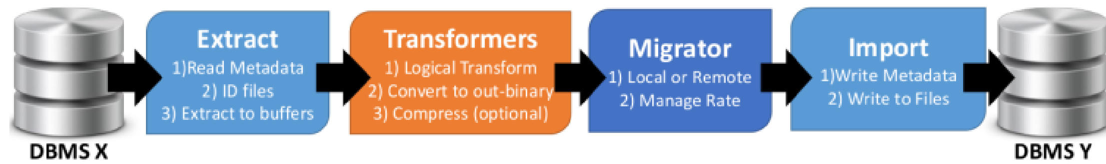


Figure 3.16: Framework Components For Portage *(Dziedzic et al., 2016)*

1. The appropriate data items must be identified and extracted from the underlying engine. Depending on the system, this may involve directly reading a binary file, accessing objects in memory, or streaming data from a distributed storage manager.

2. Data must be transformed from the source database to the destination database. This involves both, a logical transformation to change the data representation, so the destination engine can operate on the data.

3. The transformed data must be migrated between the machines. Depending on the instance locations and the infrastructure available, this step could involve the usage of pipes, sockets, files, shared memory, or remote direct memory access (RDMA)

4. The destination engine must load the data (Dziedzic *et al., 2016*).

As technology is still evolving in the domain of data processing, it can be assumed that the data migration steps from figure 3.16 may change in the near future.
In relation to the internship at Sectra, it was important to understand that the migration of patient data consists of various steps which need to be taken into consideration while creating the mismatch estimation tool. However, a profound understanding of each step was not necessary as the tool would not influence any of these steps.

3.8 Integrating the Health Enterprises (IHE)

In 1998 the IHE was created to "promote the use of standards to achieve interoperability of health information technology (HIT) systems and the effective use of electronic health records (EHRs)". The IHE community consists of volunteers from various healthcare professions who all aim for the goal of achieving consensual standard solutions.

The IHE can be seen as a council, promoting the interoperability between all HIT. Thus, the IHE analyses issues and possible solutions regarding software integrations on clinical applications by providing best-practice examples. In addition, the IHE offers guides for configuring and installing their recommended standards by providing the best workflow options. The use of internationally recognized standards such as HL7 and DICOM are always part of the guidelines for specific clinical needs in relation of optimal patient care. Overall, a system developed via IHE is able to communicate better with another system created according IHE. These systems tend to be easier to be implemented and offer a more effective data exchange (Integrating the Healthcare Enterprise, 2021) .

As the IHE guidelines appear to have a major role in today's medical environment, it is highly important to use the same terminology as indicated in the guides.

The following terminologies can be found in the implementation guides published by the IHE:

Profile: Set of functional components from a given HIT that describe its interactions and which are based on a transaction set that defines the actors and functions for each transaction.

Actor: Usually represents the HIT, or some of its components, that generates, exchanges or manages clinical data (Volume 2, IHE RAD TF-2, Transactions, 2020).

Transaction: The process that will be activated, which could have several interactions (for example, we define four actions in transaction RAD-2 (Placer Order Management): order request, order cancel, order update, and cancellation of an examination in progress) (R. da C. Oliveira, 2016).

Interaction: The tiniest part of the transaction that determines the event triggered.

An example of the used terminology can be found in figure 3.17.

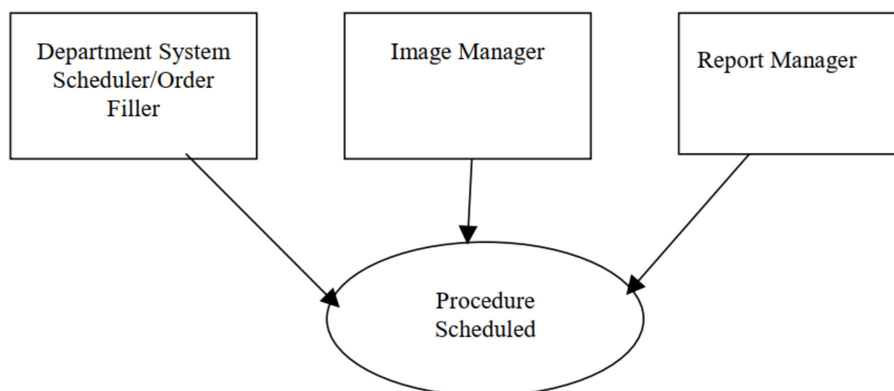


Figure 3.17: RAD-4 User Case Roles (Volume 2, IHE RAD TF-2, Transactions, 2020)

Figure 3.17 represents the so called “Procedure Scheduled” based on the IHE Technical Framework to RAD-4 (Integrating the Healthcare Enterprise, 2014).

As can be observed, there are 3 different actors:

- The Image Manager (PACS system)
- The Report Manager (RIS system)
- The Order Filler (Department System Scheduler which is normally the RIS system)

For each actor there is a well defined role:

- The “Order Filler” is in charge of scheduling procedures (order requests/examinations) and keeping its records up to date (exam update, cancellations, etc.)

- After receiving the details from the “Order Filler”, the “Image Manager” engages with the image processing.

- After receiving the details from the “Order Filler”, the “Report Manager” can use the information to handle the reports.

As can be seen, the usage of the pairs of actors/roles, gives the possibility to recreate a hospital-like environment and illustrates the interactions that should occur between two or more entities.
Additionally, to give the vendors and developers the possibility to test their system implementations against IHE profiles, annual “Connectathons” are organized by the IHE.
If a profile has been successfully tested and applied in real-world scenarios, it is added to the department’s Technical Framework, providing HIT managers and developers with standard solutions that have been proven to fix established interoperability issues (see Figure 3.18).

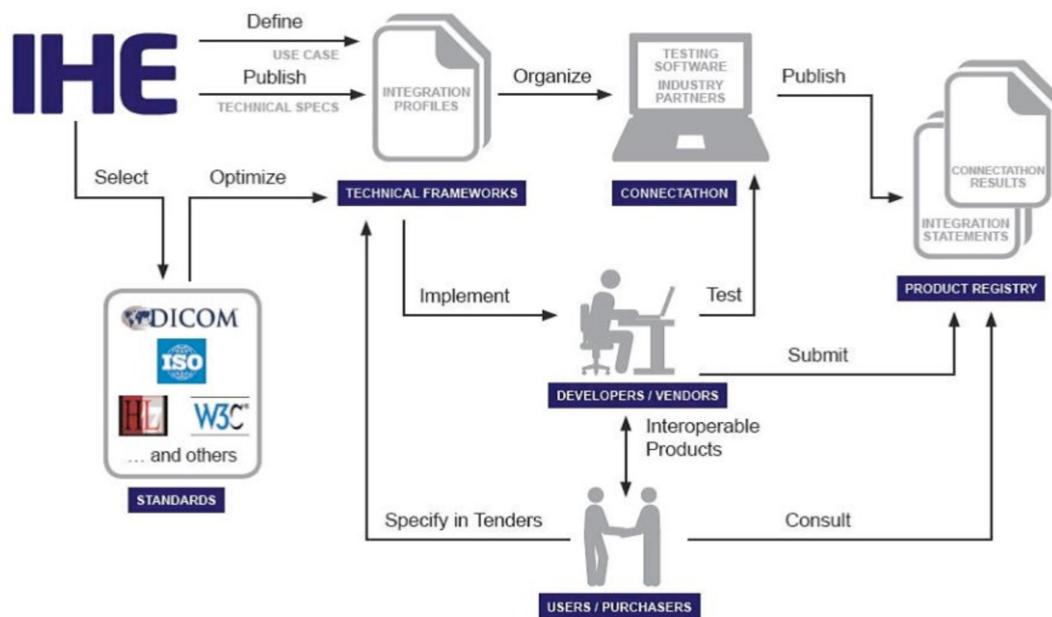


Figure 3.18: IHE Technical Framework Development Process (Integrating the Healthcare Enterprise, 2021)

When requesting proposals from vendors, IHE profile conformance can be defined as a prerequisite. When vendors are certified to be compliant and have a successfully implemented IHE profiles on their products, they can also publish their IHE Integrations Statements.

The key priority of IHE is to ensure that all patient information and data required for care are accurate and readily accessible when needed (Integrating the Healthcare Enterprise, 2021).

Regarding the allocated tasks for the internship at Sectra, it was important to follow the instructions and recommendations provided by IHE. Since IHE frameworks provide us with the most up to date conformances which should be taken into consideration on healthcare scenarios, it was necessary to create the mismatch examination tool accordingly.

3.9 Sectra

Sectra stands for “Secured Transmission” and is a Swedish company which offers cutting-edge solutions in the areas of medical IT and cyber security.

The company has more than 30 years of experience in their field and has reached by now around 2,000 installations worldwide. Sectra has become a leading global provider of imaging IT solutions that support healthcare in achieving patient-centric care. Sectra offers an enterprise imaging solution comprising PACS for imaging-intense departments (radiology, pathology, cardiology, orthopaedics) and share as well as collaborate solutions. For numerous consecutive years, Sectra has been awarded various ‘Best in KLAS’ titles for customer satisfaction and for their PACS product (SECTRA, 2021).

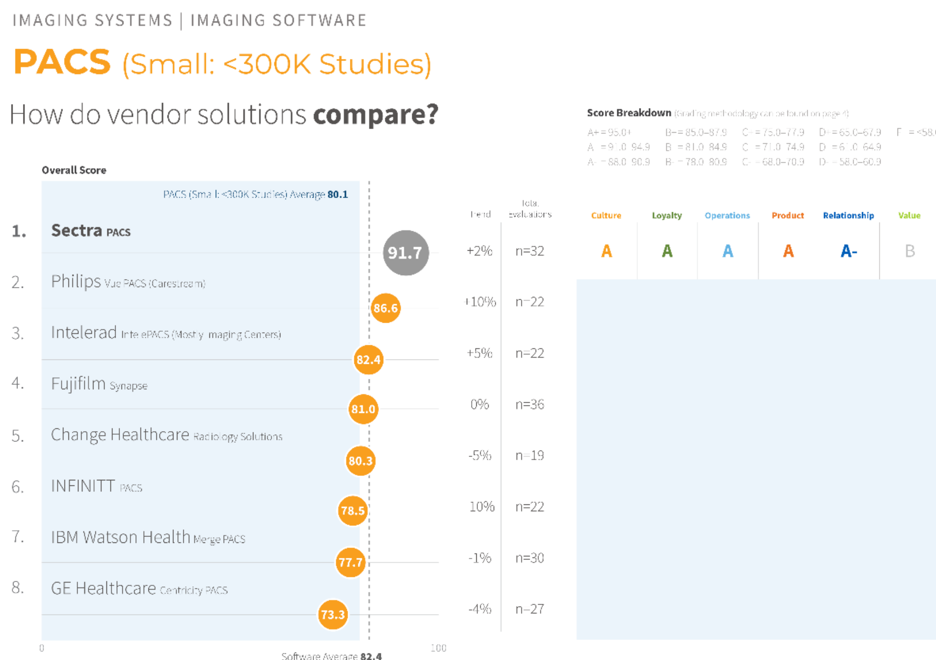


Figure 3.19: 2021 Best in KLAS Ranking (Sectra, 2021)

Apart from the PACS, the company also offers an innovative product regarding digital pathology. In this context, glass slides are being converted into digital images via digital pathology scanning solutions. The images are then generated according to the needs of the users. As a result, high resolution viewing, image analysis and interpretation can be done by the clinicians and even integrated into the PACS.

Chapter 4

Project activities

As the main goal of the internship was to create a tool for avoiding future examination mismatches during a migration process, it was important to fully understand the examination mismatch procedure. Therefore, one of the initial tasks was to do an investigation regarding patient and examination mismatch. This investigation was mainly done by gathering and by applying the information from chapter 3. As a result, a flowchart about the DICOM image mismatch procedure for the estimation tool was created.

4.1 Examination Mismatch

In general, the most common types of mismatches for files transferred into a PACS are the following:

- Medical record number (MRN) mismatches
- Date of birth (DOB) mismatches
- Patient name mismatches

If a system receives patient data which does not contain the same MRN, DOB or patient name as indicated in the database of the system, the incoming content is set as a "patient mismatch".
A possible root cause for patient identification mismatches could be the non-conformance between the various entities of a system. It is undeniable to mention that the conformity between all the used devices needs to be guaranteed. For example, some entities do not support every SOP class of a DICOM file. In this case, the images transfer is inhibited and the PACS cannot receive all the information which may lead to a patient mismatch.
Therefore, it is important to follow some guidelines regarding the integration of various entities into an archive and communication system. Nowadays, it is highly recommended to follow the guidelines based on "Integrating the Healthcare Enterprise" (IHE).

As already mentioned in the introduction part, the most common root cause of patient mis-identification is based on human error. By mistyping the patient name, DOB or MRN, chances are very high that a misidentification may occur. These types of human errors usually occur right before the patient examination. In this case, the modality user often mis-types the patient information. (Therefore, the usage of an automated mismatch estimation tool can be seen as an advantage for avoiding human errors.)

4.1.1 How images are imported into PACS:

Overall, the DICOM identifiers (accession number, patient ID, patient birthdate, patient name etc) are sent to the PACS via two different ways:

1. The identifiers can either be sent to the PACS via the local RIS. This transfer of information is done by using HL7 messages.
2. By using the DICOM Store command. In this case, the identifiers are included in the DICOM files which are sent to the PACS.

- When the DICOM storage server receives a request to store a new image, the server starts a separate thread to handle the communication. The thread takes care of the whole association and exits as soon as the association has ended.

- Just as an image arrives to the PACS, it is being pre-processed. The role of a pre-processing program is to revise the DICOM tags of the images and to verify if the images can furthermore be processed. In some cases, the pre-processing of images consists of rewriting some of the DICOM tags and even fixing images. The way an image pre-processing is being performed can vary between the different configurations.

- As soon as the incoming image passes the pre-processing, the image is being checked for a possible duplicate in the PACS. This check is done by verifying the SOP instance UID of the DICOM file (DICOM tag "8,18"). Based on the DICOM standard, 2 different images should never have the same SOP instance UID

- If the SOP Instance UID of the image is unique, the system will compare the request and the examination number indicated in the DICOM image. By default, the request number is taken from the DICOM Tag 8,50 (Accession number) and the examination number from the DICOM Tag 20,10 (Study ID). In case these tags are empty, the request and the examination number are taken from one of the following DICOM Tags:

  - 10, 20 Patient ID
  - 10, 30 Patient Birthdate
  - 10,10 Patient Name

4.1.2 DICOM image mismatch procedure:

After successfully being imported into the PACS, the DICOM files are checked for possible mismatches.

These checks are performed under the scope of DICOM Store commands which verify specific parts of the DICOM tags.

Overall, the PACS verifies if the patient ID, patient name and/or patient birthdate in the image files matches the request data in the PACS database. The name matching only verifies the characters from A-Z and 0-9.

A flowchart of regarding the matching procedure can be found below:

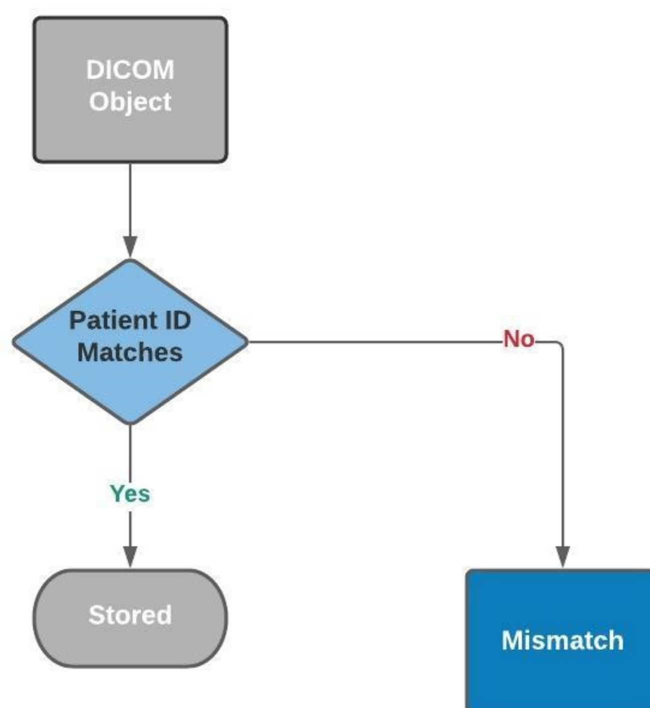


Figure 4.1: Patient ID Mismatch Flowchart

As indicated in figure 4.1, the new DICOM object will be set as “mismatch” if the patient ID of the incoming exam (DICOM file) does not match with the patient ID of the request folder from the PACS database.

The exact same mechanism is being applied for the DICOM store arguments which check the patient name and the patient birthdate.

By putting all the checks together, the following incompatibility flowchart is received:

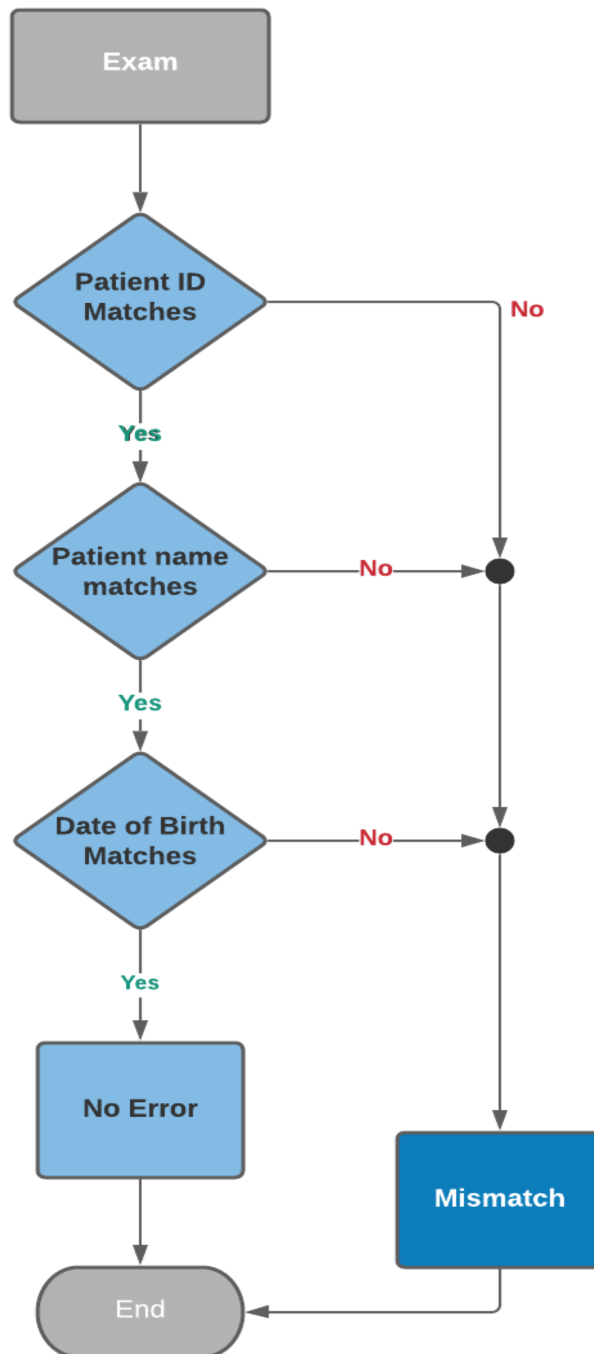


Figure 4.2: Patient ID, Patient Name, Date of Birth Mismatch Flowchart

As it may be observed in figure 4.2, the DICOM store command checks the patient ID, patient name and patient date of birth of the DICOM image. If the data does not correspond with the data in the PACS database, the DICOM file will be set as “mismatch”. The mismatch occurs as soon as one of the three arguments does not correspond with the content of the database.

If there is a mismatch, the following will happen:

- A new request number will be used (usually by adding an extra letter to the request number). Then the exact same checks will be performed as in figure 18.
- The new request will only use the content from the DICOM file. This means that the information from the RIS will not be validated (see figure 19 below).

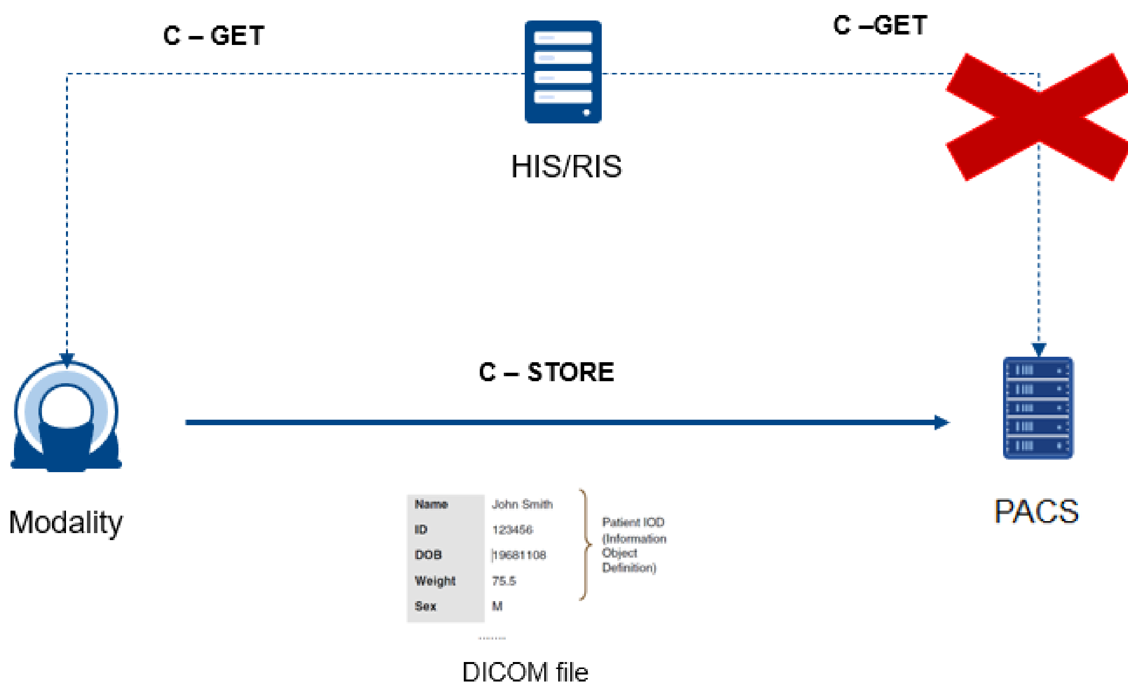


Figure 4.3: Workflow After First Mismatch

As can be seen in figure 4.3, the PACS will only register the DICOM tags from the incoming DICOM file. This procedure only appears after being registered the first examination of incompatibility.
PACS users will receive a notification on the user interface indicating that the examination is a mismatch examination. These mismatches can be manually corrected by the PACS admin.

As the flowchart in Figure 4.2 served as the basis for the mismatch estimation tool, it was important to fully understand its purpose before creating the tool.

4.2 Methodology of the project

Based on the flowchart from figure 4.2, the following task was to become more familiar with the applications and procedures used for the creation of the mismatch estimation tool. In this context, it was essential to get a deeper understanding regarding SQL, Windows Powershell and the creation of IT related tools by using the procedures from IHE.

4.2.1 Structured Query Language (SQL)

The initial purpose of SQL is to interact with relational databases. This means that the SQL allows its users to create new databases, insert new data to them, manage the data in them and also retrieve the data from the databases. SQL was developed in the 1970s by IBM and has ever since advanced in its importance. Nowadays, this query language is seen as a standard by the International Standards Organization (ISO).

By now, this language is not only used by database administrators, but also by different types of professionals such as data analysts and software developers.

Today, there exist different types of database management systems (DBMS) which use the SQL framework. A DBMS is a software package designed to define, manipulate, retrieve and manage data in a database. A DBMS generally manipulates the data itself, the data format, field names, record structure and file structure. It also defines rules to validate and manipulate this data.

Some commonly known DBMS's are:

- MySQL
- SQL Server
- Oracle
- dBASE
- PostgreSQL

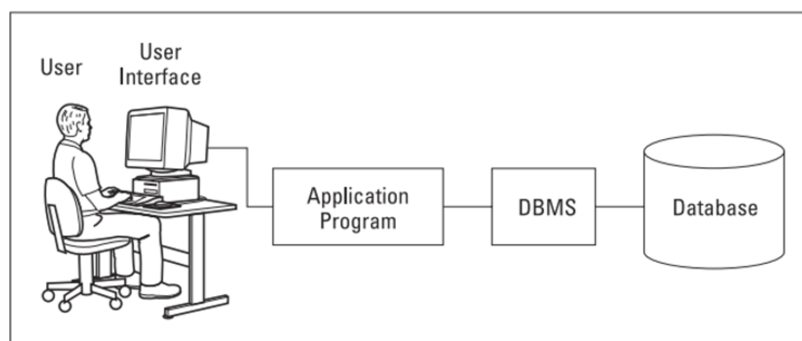


Figure 4.4: Block Diagram of DBMS Based Information System

Figure 4.4 illustrates a schematic overview of a DBMS-based information system. As can be seen, to successfully retrieve data from the database, the user first needs to connect to a DBMS.
Regarding the task at SECTRA, the DBMS used for developing the estimation tool was “PostgreSQL”. The reason for choosing PostgreSQL was cost related. As this type of DBMS can be downloaded and installed for free, it appears to be an inexpensive option for potentially new PACS customers.

SQL commands:

The queries and additional SQL operations are inserted as statements. The most common SQL statements are the following: select, add, insert, update, create, alter, truncate and delete.
All commands of SQL fall in one of the following categories:

1. DDL (Data Definition Language)
These statements are used to create tables and databases and define field properties or table properties. Examples of commands that fall in this category are CREATE,ALTER and DROP statements

2. DML (Data Manipulation Language)
The statements that falls under this category are used to update data or add or remove data from tables. UPDATE, DELETE and INSERT commands fall under this category.

3. DCL (Data Control Language)
It is used to control who access the data. The commands that come under this category are GRANT and REVOKE

4. TCL (Transaction Control Language)
This language is used to commit data and restore data. COMMIT and ROLLBACK falls under this category.

5. DQL (Data Query Language)
This is to retrieve data from SQL server. SELECT statement falls in this category.

To successfully create the estimation tool, a profound knowledge of all the different languages was required.

4.2.2 Powershell

Powershell (also known as Windows PowerShell and PowerShell Core) is a cross-platform framework from Microsoft. It is used for the automation, configuration and management of computer systems by using a command line interpreter and a scripting language.
Windows PowerShell is based on the Common Language Runtime (CLR) of the .NET Framework and is supplied with Windows as part of the Windows Management Framework (WMF) (Microsoft, 2021). Powershell comes with a command-line interface which works in an interactive way. The commands can be entered and the resulting output is displayed to the user. On the contrary to the regular command line, Powershell offers the possibility to create entire scripts for managing applications but also the entire operating systems.
All in one, Windows Powershell can be applied in four kinds of so called “commands” (Microsoft, 2021):

- Cmdlets
- Powershell functions
- Standalone executable programs
- Scripts

An example of a Powershell command-line can be found below:

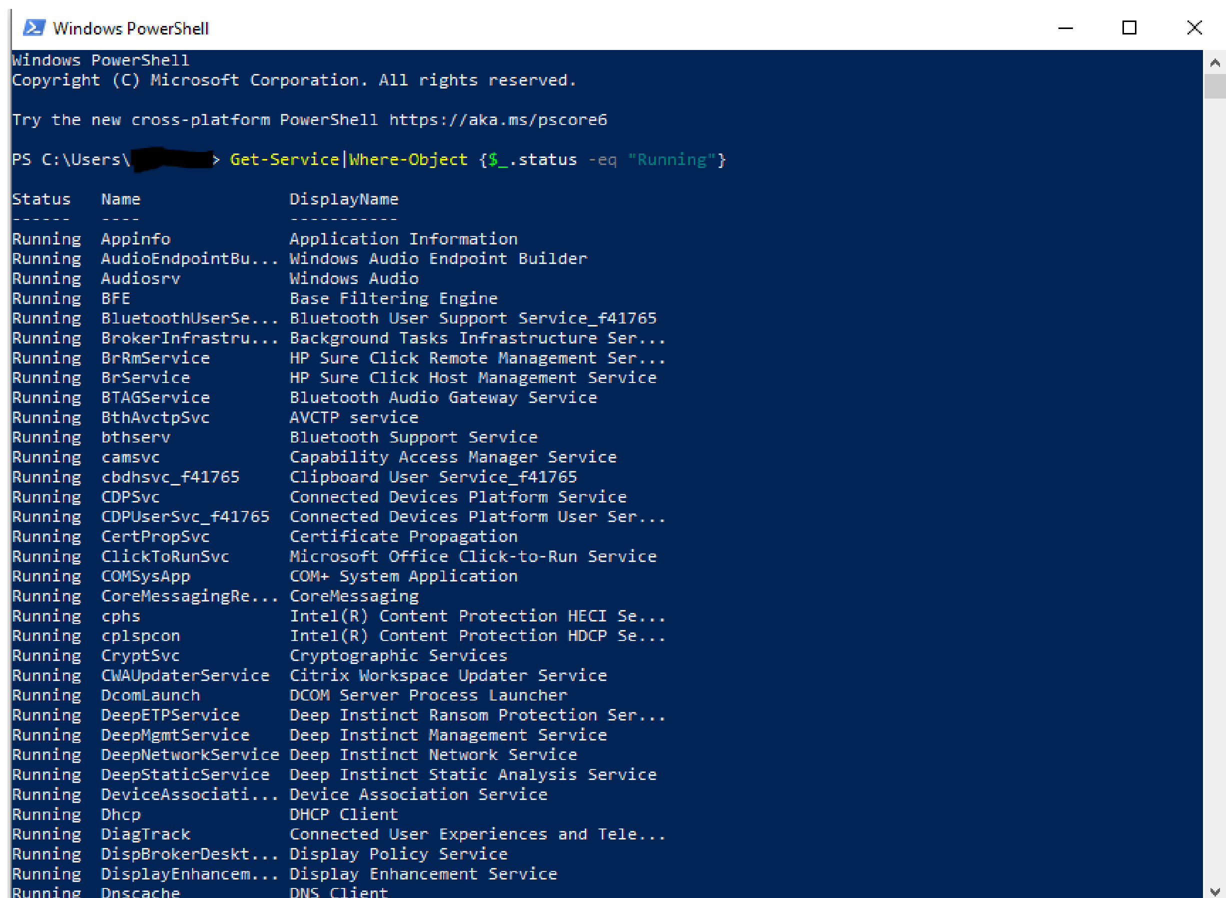


Figure 4.5: Powershell Commandline

4.2.3 Image acquisition workflow according to IHE standards:

To be able to properly understand the workflow of a PACS image acquisition according to IHE, one of the initial tasks was to create a general scheme by using the IHE terminology. This research was made by using the following guide: *IHE Radiology Technical Framework, Volume 2 (RAD TF-2): Transactions (Volume 2, IHE RAD TF-2, Transactions, 2020)*.

To fully understand the instructions and explanations of the guide, a proficient understanding of HL7 and DICOM was required.
By studying the IHE guidelines, the following conclusion was made:
To successfully perform a basic image acquisition to a PACS, the following entities need to be involved in the process:

1. Scheduling the exams
2. Querying the modality worklist
3. Storing the images

These three steps correspond to the following IHE transactions:

1. Procedure Scheduled [RAD-4]
2. Query Modality Worklist [RAD-5]
3. Creator Images Stored [RAD-18]

A deeper explanation about each of the mentioned IHE transaction is indicated below.

Procedure Scheduled [RAD-4]:

The RAD-4 transaction indicates that a message from the Department System/Order Filler is sent to the Image Manager and the Report Manager informing the entities that a procedure was scheduled. A schematic representation can be found below:

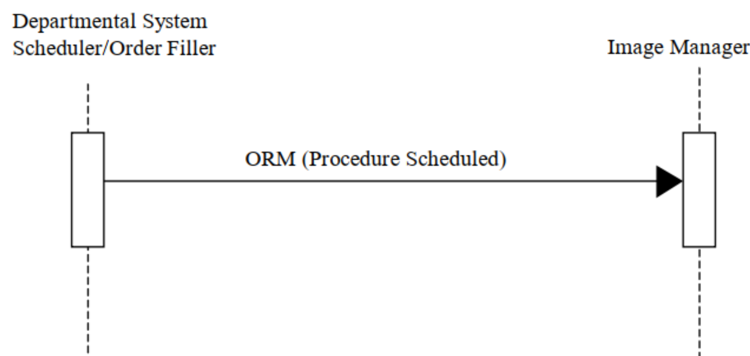


Figure 4.6: Interaction Between Two Entities Implementing HL7 V2.3 (Volume 2, IHE RAD TF-2, Transactions, 2020)

Usually, the RAD-4 transaction is used to trigger or to change scheduled orders. In present work the "Order Filler" is normally the RIS and the Image Manager is the PACS. The communication between the entities is performed by the ORM^001 message type.

Table 4.1: ORM message structure (Volume 2, IHE RAD TF-2, Transactions, 2020)

Segment	Description	Chapter in HL7 v2.5 documentation
MSH	Message Header	Chapter 2
PID	Patient Identification	Chapter 3
PV1	Patient Visit	Chapter 3
{ORC	Common Order	Chapter 4
OBR}	Observation Request	Chapter 4

As indicated in table 4.1, the ORM message structure requires a minimum of five HL7- segments: MSH, PID, PV1, ORC and OBR.

Query Modality Worklist [RAD-5]:

According to IHE, the scope of this transaction should be divided into two circumstances. One of the circumstances is for planning an accession. The second circumstance is for planning an import of "Evidence Objects" or "Hardcopy".

This transaction appears at the Acquisition Modality, more precisely at the acquisition/scan performed by a technologist.

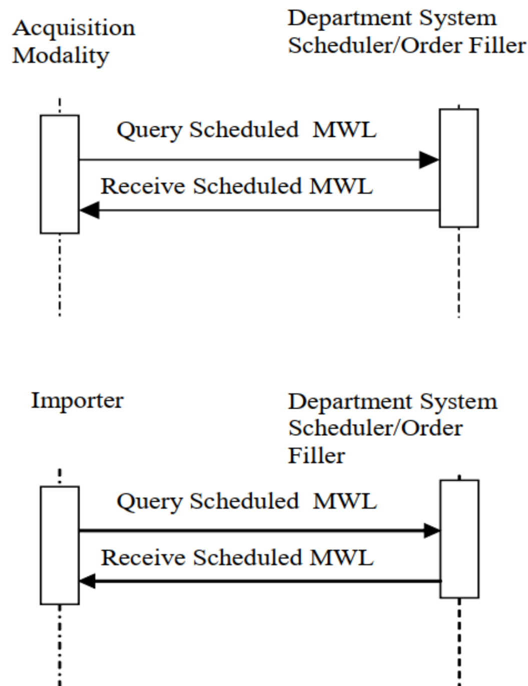


Figure 4.7: Interaction Diagram (Volume 2, IHE RAD TF-2, Transactions, 2020)

The transaction between the acquisition modality and the department system scheduler is initiated by a C-FIND Request of the DICOM modality. In this case, the Acquisition Modality appears to be the SCU and the Order Filler takes the SCP role. According to IHE, the Acquisition Modalities should support at least one of the following key combinations:

1. For a patient based query, the Acquisition Modality shall query for a worklist specific for a patient by using the DICOM tags from table 2:

Table 4.2: MLW Keys for Query by Patient (Volume 2, IHE RAD TF-2, Transactions, 2020)

Matching Key Attributes	Tag
Patient's Name	(0010,0010)
Patient ID	(0010,0020)
Accession Number	(0008,0050)
Requested Procedure ID	(0040,1001)

2. For querying a broad worklist, the SCU needs to support all combinations of the matching keys indicated below:

Table 4.3: MWL Keys for Query by Patient (Volume 2, IHE RAD TF-2, Transactions, 2020)

Matching Key Attributes	Tag
Scheduled Procedure Step Start Date	(0040,0002)
Modality	(0008,0060)
Scheduled Station AE Title	(0040,0001)

Creator Images Stored [RAD-18]:

In this transaction type, the Evidence Creator transfers the recently created images to the Image Archive.

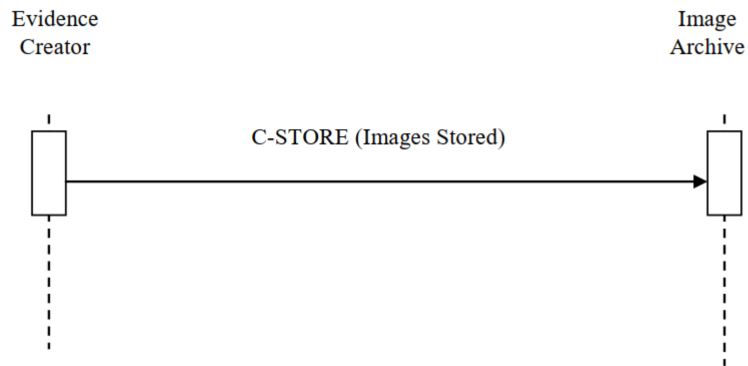


Figure 4.8: Interaction Diagram (Volume 2, IHE RAD TF-2, Transactions, 2020)

As indicated in figure 4.8, the Evidence Creator uses the DICOM C-Store message for transferring images. In this case, the Evidence Creator appears to be the DICOM Storage SCU and the Image Archive is the DICOM Storage SCP.

The DICOM objects should be stored in a way so that they can be retrieved again by another SCU.

Based on the previously mentioned transaction types, the following illustration was made:

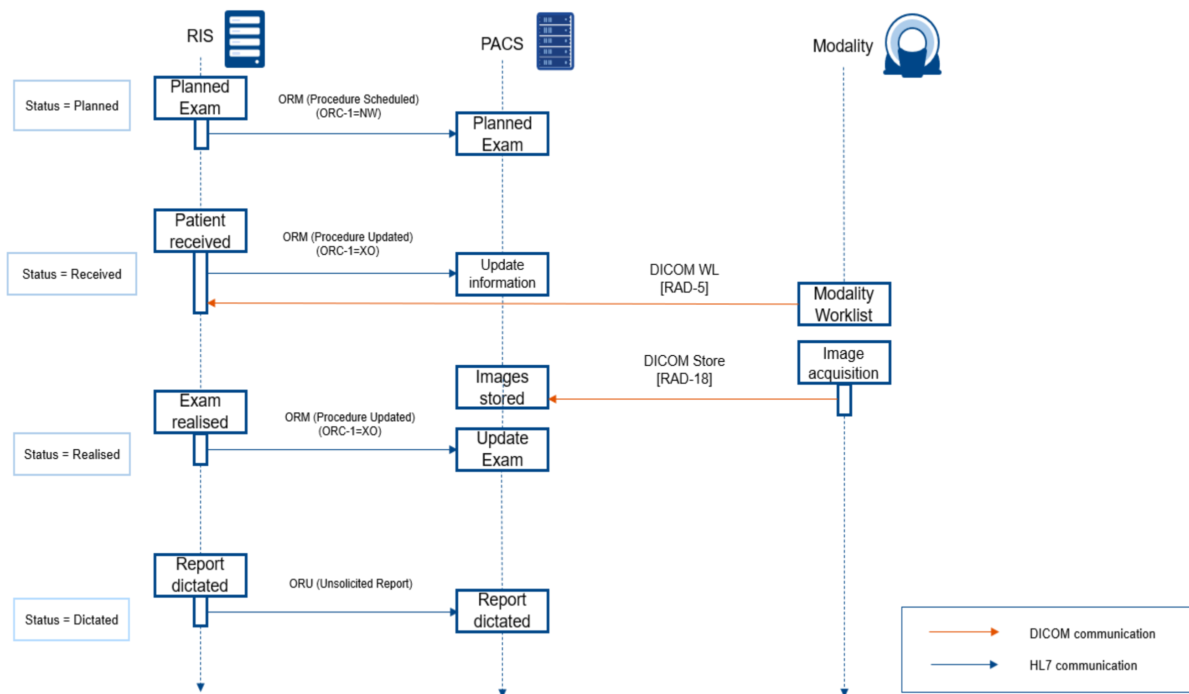


Figure 4.9: Basic Image Acquisition Workflow

As can be observed, the transaction types: RAD-4, RAD-5 and RAD-18 were combined into one model. The representation of image 4.9 is based on the guidelines from: *IHE Radiology Technical Framework, Volume 2 (RAD TF-2): Transactions*. Therefore, its content should be conform with IHE.

4.2.4 Creating the mismatch estimation tool:

Based on the requirements of the mismatch estimation tool which was illustrated in 1.2*: Purpose and main goal of the task,* the following flowchart was created (See Appendix for bigger image):

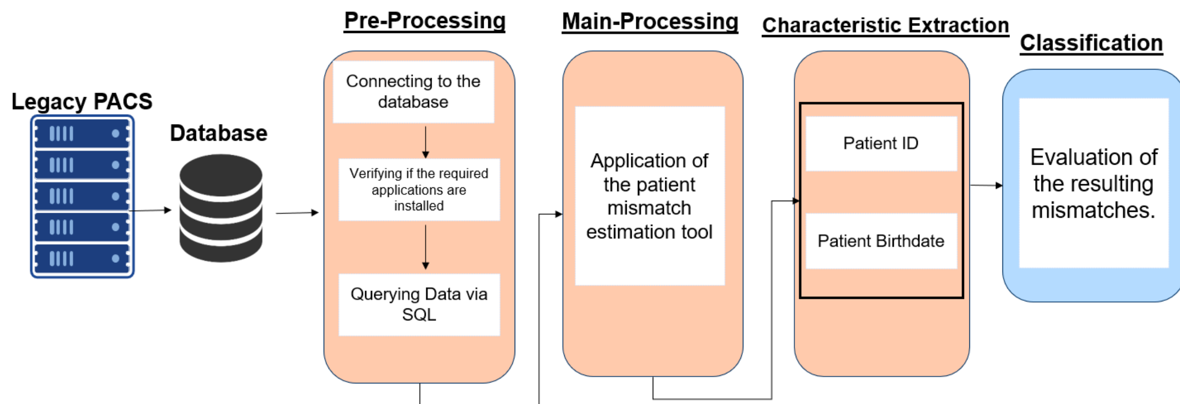


Figure 4.10: Flowchart Of The Mismatch Estimation Tool

As can be seen, the procedure of the estimation tool can be subdivided into four main phases:

1. Pre-processing of patient data
2. Main-Processing of patient data
3. Characteristic Extraction of patient data
4. Classification of the output.

An illustration of each phase can be found below:

Pre-Processing:

To successfully fetch the data from the legacy PACS, the machine which uses the estimation tool, will first have to connect to this legacy PACS.

Connecting to the database:

The connection towards the legacy PACS is executed through a remote desktop software. Depending on the customer for the migration process, different remote desktop applications can be used. Overall, a virtual private network (VPN) is used to connect to the legacy PACS.

As soon as the connection is established, the following check will run.

Verifying if the required applications are installed

To successfully apply the patient mismatch estimation tool, “psqlODBC” and “Importexcel” need to be installed. This verification is done by using Powershell under the following script:

```
######### Check if psqlODBC and importexcel are installed #########

$software = "psqlODBC_x64";
$installed = (Get-ItemProperty
HKLM:\Software\Microsoft\Windows\CurrentVersion\Uninstall\* |
Where { $_.DisplayName -eq $software }) -ne $null

If(-Not $installed) {
    Write-Error "'$software' is NOT installed. Please install
it before continue."
    exit
}
if (-Not (Get-Module -ListAvailable -Name "importexcel")) {
    Write-host "Module import-excel does not exist. Please
install it before continue."
    exit
}
##################################################################
```

In case the script detects that none of the two applications is installed, the estimation tool will not run. In this case, the user will be asked to install “psqlODBC” and “Importexcel” first.

Querying Data via SQL:

If the applications “psqlODBC” and “Importexcel” are correctly installed, the data can be queried from the database of the legacy PACS.

One example of such an SQL query can be found below:

```
########### Studies with mismatch on Patient BirthDate ###########
SELECT
`"PatientID`" as `"PatientID`",
`"PatientName`" as `"PatientName`",
`"PatientBirthDate`" as `"PatientBirthDate`",
`"AccessionNumber`" as `"AccessionNumber`",
`"StudyId`" as `"StudyId`",
coalesce(`"StudyDescription`", '') as `"StudyDescription`",
coalesce(`"BodyPart`", '') as `"BodyPart`",
coalesce(`"StudyDate`", '') as `"StudyDate`",
`"NumberofStudyRelatedInstances`",
`"StudyInstanceUid`",
coalesce(`"Modality`", '') as `"Modality`",
coalesce(`"StationName`", '') as `"StationName`"
from `"ExamstoMove`"
where `"PatientID`" in
(
SELECT `"PatientID`" FROM `"ExamstoMove`"
GROUP BY `"PatientID`"
having count(distinct `"PatientBirthDate`")>1);
################################################################
```

As can be seen, the output of this query provides all the studies of the database where the birthdate of the same patient is not matching. A full list with all the various SQL queries can be found in the Appendix section.

Main Processing:

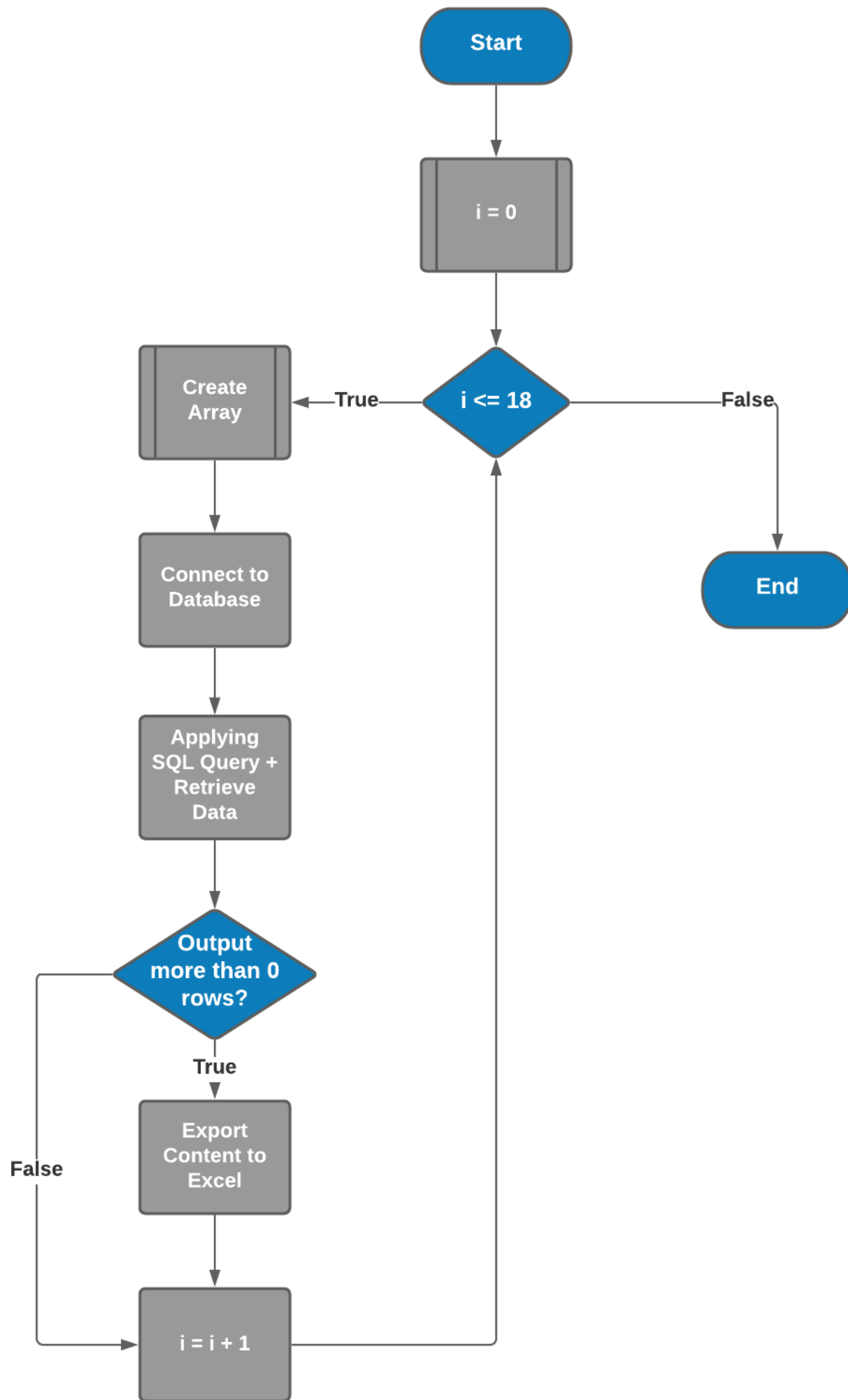


Figure 4.11: Flowchart of The Main Processing

Figure 4.11 contains the flowchart of the main process for the estimation tool. The number i represents the number of the query used for retrieving the data from the legacy PACS. In this example 19 different queries are being applied. Therefore, the condition i <= 19 was set.

For each query, the Powershell script will connect to the database, apply the SQL query and retrieve the information. If the output contains any information (more than 0 rows), its content will be exported to an Excel file. Then, the next query will be applied in the exact same way. The script will stop running as soon as all the queries were used. In this case the script would stop after 19 queries. The exported content of all the queries will be presented on one Excel file.

The entire Powershell script as well as all the SQL – queries can be found in the Appendix section.

Characteristic Extraction:

The Excel spreadsheet created during the previous process can then be customized according to the needs of the customer or the migration team.

Since Excel offers a variety of functions and applications, the user can sort the data in multiple ways. One example for sorting the content would be by using the “PivotTable” function. In this case the Excel user could have a clear overview regarding the different examination mismatches based on “Patient ID” and “Patient Birthdate”(see table 4.4 indicated below).

Table 4.4: Example of output from the estimation tool

Accession number	DOB	Patient ID	MODALITY	EXAM_STUDY_UID	Exam Date
10074214155	20/03/1995	518320	CT	2.16.840.1.113669.632.20.1496385081.537038478.10020342541	04/03/2019
10007873654	13/07/1954	393886	CR	2.16.840.1.113669.632.20.1496385081.537038478.10020342550	26/07/2020
10023433125	24/06/1967	338361	MR	2.16.840.1.113669.632.20.1496385081.537038478.10020342543	04/01/2019
10073066066	03/04/1991	343694	US	2.16.840.1.113669.632.20.1496385081.537038478.10020342556	02/09/2020
10045279996	12/05/1958	169408	US	2.16.840.1.113669.632.20.1496385081.537038478.10020347475	01/06/2019
10041966908	01/09/1983	132978	OT	2.16.840.1.113669.632.20.1496385081.537038478.10020347475	15/04/2020
10066583684	21/01/1957	339196	MR	2.16.840.1.113669.632.20.1496385081.537038478.10020347475	09/08/2019
10032942553	01/02/1956	282466	CT	2.16.840.1.113669.632.20.1496385081.537038478.10020347475	31/08/2020
10023404861	18/01/1994	552947	CT	2.16.840.1.113669.632.20.1496385081.537038478.10020347476	10/01/2019
10026953497	05/03/1958	143223	CR	2.16.840.1.113669.632.20.1496385081.537038478.10020347476	01/03/2020
10009534277	08/06/2002	743425	CR	2.16.840.1.113669.632.20.1496385081.537038478.10020347476	14/04/2019
10083558990	23/07/2002	142846	CR	2.16.840.1.113669.632.20.1496385081.537038478.10020347476	24/11/2020
10020890953	03/05/1965	489520	CR	2.16.840.1.113669.632.20.1496385081.537038478.10020347588	01/04/2020
10094798748	30/12/1989	551234	MR	2.16.840.1.113669.632.20.1496385081.537038478.10020347507	18/02/2020
10003740735	06/02/1971	667040	CT	2.16.840.1.113669.632.20.1496385081.537038478.10020347507	12/10/2019
10008895106	12/09/2004	392099	CT	2.16.840.1.113669.632.20.1496385081.537038478.10020347507	28/06/2019
10034722755	15/02/2001	649335	CT	2.16.840.1.113669.632.20.1496385081.537038478.10020347507	25/02/2019
10098633924	28/10/1988	776367	CR	2.16.840.1.113669.632.20.1496385081.537038478.10020347509	17/09/2020
10062610695	05/05/1992	756296	CR	2.16.840.1.113669.632.20.1496385081.537038478.10020347509	17/05/2019
10088514983	12/06/1986	704011	OT	2.16.840.1.113669.632.20.1496385081.537038478.10020347509	01/01/2019
10042494050	10/09/1971	365313	CR	2.16.840.1.113669.632.20.1496385081.537038478.10020347509	11/01/2019
10035703333	26/11/2001	653141	US	2.16.840.1.113669.632.20.1496385081.537038478.10020347509	29/08/2019
10053242120	22/02/1973	634908	US	2.16.840.1.113669.632.20.1496385081.537038478.10020347539	19/12/2020
...	...	...	...	... ...	...
...	...	...	...	... ...	...
Sum of mismatches:	753				

Table 4.4 represents a possible output of the estimation tool. For instance, the spreadsheet contains a list of all the examinations which would be set as “mismatched” during a PACS migration.
In this example, each examination is defined with an accession number, patient date of birth, patient ID, modality type (according to the DICOM standard), exam study UID and Exam date. The patient name was not extracted as this variable can change over time. One of the reasons for a change in patient name can be the marital status of a patient.
As mentioned previously, the spreadsheet can furthermore be edited according to the needs of the end user. In this example, the sum of all the mismatches was added by the end user.

Chapter 5

Conclusion and Future Work

5.1 Results

During the internship, the estimation tool was developed and used in a Sectra test environment only. The patient data used for the creation of the tool originated from real patients. However, all the DICOM tags which could identify these patients were changed with fictional information such as a fictional patient name and patient date of birth.

The estimation tool appeared to work as designed:

- All examinations which were supposed to be set as mismatched were successfully displayed on an Excel spreadsheet.
- All the 19 SQL queries were running successfully. The Powershell script was able to verify if PostgreSQL was installed on the hosts.
- The examinations which were not supposed to be set as mismatched were not displayed on the spreadsheet.

As the mismatch estimation tool passed all the tests needed for its approval, the tool could be used in a production-like environment. In this case, real patient data would be processed.

5.2 Future Work

Apart from the fact that the estimation tool still needs to be tested on real patient data coming from a real legacy PACS, there are further actions which could be taken:

- As for now, the estimation tool only provides an excel sheet containing a list of possible mismatches which would occur during the migration process. This means that the user still needs to do the estimation regarding the duration of the entire migration process by himself/herself. By knowing how long the migration of a single examination takes and by knowing how much time the fixation of a single examination mismatch takes, the estimation tool could display a first impression regarding the duration of the entire PACS migration.

- Automatically fixing the enlisted mismatches. An additional upgrade for the estimation tool would be to automatically solve examination mismatches. This process would not only decrease the duration of the migration drastically, but it would also take away the task of correcting the mismatches from the migration team. However, such an automation process requires machine learning which would significantly increase the duration for developing such an application.
- Integrating macros in Excel. As the users of the estimation tool need to edit the Excel file manually, it would be helpful for them to have some macros already integrated.

5.3 Conclusion

Despite the latest advancements in medical IT, the procedures of matching patient data during a migration process still needs to be improved. Such an improvement does not only save financial costs, but it also creates a safer environment for the patients.
The scope of the internship at Sectra was to create a tool which would estimate the number of patient examination mismatches during a PACS migration process. To fully understand the mechanism of the tool, a profound understanding regarding: DICOM, HL7, PACS as well as the general workflow of a radiology department were required. Additionally, a deep understanding of the recommendations provided by the IHE was needed, so that the tool would stay conform with all the IHE requirements for the radiology department. To demonstrate that the recommendations of the IHE were taken into consideration, a graphical representation of a general image acquisition workflow was created (see figure 4.9).
The mismatch estimation tool was created by using PostgreSQL for querying patient data from the legacy PACS and by using *Windows Powershell* for executing the tool.

While creating the tool, two main challenges had to be taken care of:

1. One challenge was to fully understand the mechanism behind the examination mismatches. As the mechanism of examination mismatches was the foundation of the tool, it was necessary to demonstrate a deep understanding of it. Therefore, an appropriate flowchart was created (see figure 4.2).

2. The second major challenge was the development process of the tool. Based on the information of the flowchart, the tool had to be created in a way so that it would follow the given procedure. Hence, this task required additional research regarding scripting and execution.

The final results of the tool appeared to be promising as it provided the requested list of all the possible mismatches which would take place during a migration process. However, the tool was only used in a lab environment containing familiar patient data. This means that the tool could possibly fail during production as it would process unfamiliar patient data and be used in an environment which would possibly be different from the lab.
Even though the tool was working as designed, some additional features could be added. These features would not only make the tool more user friendly, but also to provide more valuable information to the users. As these additional features require knowledge in machine learning and a deeper understanding in scripting as well as in Microsoft Excel, some additional time for the development should be taken into consideration.
The fact that the internship took place during the COVID19 pandemic has added some additional challenges to it. As the company is providing technical support to Iberian hospitals, this service had to be prioritized during the peak of the pandemic. As a result, the supervision of the tasks had to be delayed and the duration of the internship was extended.
All in one, it can be said that the internship at Sectra covered multiple aspects of Imagiology which were introduced during the Masters degree in Biomedical Engineering.

The accomplished tasks provided a deeper understanding about the technology used in the radiology department and about the workflow regarding image acquisition. Furthermore, the coding and scripting skills taught during the Masters degree have evolved with the creation of the mismatch estimation tool.

Appendix A

Mismatch estimation tool

This section contains the Windows Powershell script used for the examination mismatch estimation tool. The script also contains all SQL queries used for retrieving patient data from the legacy PACS. In addition, some flowcharts which were created during the internship as well as the internship proposal provided by Sectra can be found in this section.

A.1 Sectra Internship Proposal

Student internship proposal

Títle:	Patient demographics mismatch estimator
Company:	Sectra (Artces Unipessoal, Lda.)
Tutor:	Pedro Miguel Rodrigues (prodrigues@porto.ucp.pt)
Co-tutor:	Daniel Carvalho (daniel.carvalho@sectra.com)
Applicant:	Randy Quintus
Applicant profile:	Basic knowledge on IT
Remuneration:	Food allowance

Project summary

Patient demographics mismatch is a mechanism that Sectra PACS (medical imaging archive software) implements, in order to identify and avoid potential risk of placing medical examinations in the wrong patient.

When medical images arrive to Sectra PACS, their patient data is compared against the known information about the patient demographics, like the patient name, patient birth date and patient sex. If there is a mismatch, then the examination is placed in a separate list, and is not shown to the clinicians, until someone verifies the patient identity.

In cases where there are legacy PACS systems being migrated to Sectra PACS, this patient demographics mismatch could be raised thousands of times, but then the PACS responsibles would have a number of mismatches to solve that is greater than their actual capacity.

In order to avoid this, we would like to create a tool, that would compare the patient demographics in exams from legacy systems with a set of reference data, providing the required statistics about mismatches, before this information is actually sent to the Sectra PACS and real patient data mismatches are raised. This way, the project team (either Sectra's or the Hospital's) can take a decision on how to cope with them.

Expected outcome

Being able to connect to two databases, one with legacy patient demographics, the other with reference patient demographics, a simple tool is to be developed, that can mimic the Sectra PACS patient demographics mismatch mechanism, and issue statistics about it.

- The number of estimated patient name mismatches
- The number of estimated patient birth date mismatches
- The number of estimated patient sex mismatches

Areas of study

- Sectra PACS administration manuals (how the patient mismatch mechanisms work)
- Databases (simple queries in Microsoft SQL server and PostgreSQL)
- Programming languages (Python, Javascript, but might be others)
- Concepts on healthcare metadata related to patients and examinations
- The DICOM standard

Work plan

ID	Activity	Duration (days)
1	Investigation period on the areas of study	10
2	Implementation of the mismatch estimation tool	12
3	Tests with the tool and data sets	4
4	Improvements on the tool	8
5	Documentation about the tool and processes	6
	Total	**40**

Infrastructure to use

Existing Sectra Iberia laboratory (virtual machines for develop and test purposes)

The GIT code repository, where any source code and documentation should be stored

Work location

In the Sectra Iberia Porto offices.

References

System Administrator's Guide - ImageServer/s - Sectra PACS, Version 21.2, September 2019 – Section 8.4.5 "Mismatch" - https://userweb.sectra.se/imtec/pacsproddocweb.nsf/F0E3E1DFF1F2D73EC12584890033ADD2/$FILE/sysadm_guide_ISs.pdf/sysadm_guide_ISs.pdf

The DICOM Standard - https://www.dicomstandard.org/current/

The SQL Language in PostgreSQL - https://www.postgresql.org/docs/9.4/sql.html

A.2 Powershell Script containing SQL queries for the patient data migration

<#

.SYNOPSIS

Migration_DB_SanityChecks

.DESCRIPTION

Migration_DB_SanityChecks goal is to automatically create the excel file that needs to be delivered with MigrationTestRecord

To execute this script you must have psqlODBC_x64 and import-excel installed.

To install psqlODBC_x64 go to https://www.postgresql.org/ftp/odbc/versions/msi/ and download the lastest verion and install it.

To install import-excel you must have internet access, download the importexcel from the "https://files.sectra.se/index.php/s/e34iXnK3PkQBMJt" and unzip the folder to C:\Program Files\WindowsPowerShell\Modules

Open powershell as admin and run "Install-Module ImportExcel -scope CurrentUser" and follow the instructions shown on the powershell.

.PARAMETER dbServer

DB Server (either IP or hostname), default value is localhost

.PARAMETER dbName

Name of the database, default value is mirthmover

.PARAMETER dbUser

User we'll use to connect to the database/server, default value is mirth

.PARAMETER dbPass

Password for the dbUser, default value is 1234

.PARAMETER cutofdate

Cutofdate, default value is 20050101

.PARAMETER pid_pattern

Pattern used on PID, if multiples are applied use | (or) between them, default value is [0-9]%

.PARAMETER accno_pattern

Pattern used on ACCNO, if multiples are applied use | (or) between them, default value is [0-9]%

.PARAMETER xlfile

Excel filename, default value is SanityChecks.xlsx

.Notes

Use at your own risk

version 1 - 26/10/2020

.EXAMPLE

Migration_DB_SanityChecks.ps1

If you update all the values on the script you can run just the Migration_status_lists.ps1

.EXAMPLE

Migration_DB_SanityChecks.ps1 -dbServer "REMOTE" -dbName "MirthMover" -dbPass "1234" -cutofdate "20100101"

Just updating the values that are different from the default

#>

```
param(
    [string]$dbServer     = "localhost" # DB Server (either IP or
hostname)
    [string]$dbName      = "MirthMover", # Name of the database
    [string]$dbUser             ,   # User we'll use to connect to the
database/server
    [string]$dbPass            ,   # Password for the $dbUser
    [string]$cutofdate  = "20060101",  # Cutofdate
    [string]$pid_pattern    = "RM3[0-9]{1,8}", # Pattern used on PID,
if multiples are applied use | (or( between them
    [string]$accno_pattern  = "RM3[0-9]{1,8}" # Pattern used on ACCNO,
if multiples are applied use | (or) between them
    [string]$xlfile           = "SanityChecks.xlsx")  # Excel filename

-- Every time the following function is called, a new line on the
$dataTable will be added ----

Function New-line {
param($PatientID,$PatientName,$PatientDOB,$AccessionNumber,$StudyId,
$StudyDescription,$BodyPart,$StudyDate,$NumberofStudyRelatedInstance
s,$StudyInstanceUid,$Modality,$StationName)
    $row=$dataTable.NewRow()
    $row["PatientID"]=$PatientID
    $row["PatientName"]=$PatientName
    $row["PatientDOB"]=$PatientDOB
    $row["AccessionNumber"]=$AccessionNumber
    $row["StudyId"]=$StudyId
    $row["StudyDescription"]=$StudyDescription
    $row["BodyPart"]=$BodyPart
    $row["StudyDate"]=$StudyDate

$row["NumberofStudyRelatedInstances"]=$NumberofStudyRelatedInstances
    $row["StudyInstanceUid"]=$StudyInstanceUid
    $row["Modality"]=$Modality
    $row["StationName"]=$StationName
    $dataTable.Rows.Add($row)
}
```

```
---- The following function applies the inserted query and exports
the result into and excel file ---

Function LIST_QUERY()
{
param( [string]$query,
    [string]$WorksheetName,
    [string]$TableName )
 $output = @()   # initializes the variable for the DB connection.#
The @ indicates an array. @() simply creates an empty array.
 $conn = New-Object System.Data.Odbc.OdbcConnection # added the db
information to connect
 $conn.ConnectionString             =            "Driver={PostgreSQL
Unicode(x64)};Server=$dbServer;Port=5432;Database=$dbName;Uid=$dbUse
r;Pwd=$dbPass;"
 $conn.Open()   # initializes the variable for the query
 $cmd = New-object System.Data.Odbc.OdbcCommand($query,$conn)
 $cmd.CommandTimeout = 0 # initialize the variable for receiving the
records from the DB
 $ds = New-Object system.Data.DataSet # executed the query and fill
the $ds with the records returned by the DB
 (New-Object system.Data.odbc.odbcDataAdapter($cmd)).fill($ds) | out-
null $conn.close()
# if the rows are more than 0 we want to save otherwise not
    if ( ($($ds.Tables[0].Rows).count) -gt 0)
    {
# created a kind of a class to add the rows and then export to the
excel without strange columns
    $dataTable = New-Object System.Data.DataTable("DataSet")
    $dataTable.Columns.Add((New-Object        System.Data.DataColumn
"PatientID"))
    $dataTable.Columns.Add((New-Object        System.Data.DataColumn
"PatientName"))
    $dataTable.Columns.Add((New-Object        System.Data.DataColumn
"PatientDOB"))
    $dataTable.Columns.Add((New-Object        System.Data.DataColumn
"AccessionNumber"))
    $dataTable.Columns.Add((New-Object        System.Data.DataColumn
"StudyId"))
    $dataTable.Columns.Add((New-Object        System.Data.DataColumn
"StudyDescription"))
    $dataTable.Columns.Add((New-Object        System.Data.DataColumn
"BodyPart"))
    $dataTable.Columns.Add((New-Object        System.Data.DataColumn
"StudyDate"))
    $dataTable.Columns.Add((New-Object System.Data.DataColumn
                           "NumberofStudyRelatedInstances"))

 $dataTable.Columns.Add((New-Object           System.Data.DataColumn
"StudyInstanceUid"))
    $dataTable.Columns.Add((New-Object        System.Data.DataColumn
"Modality"))
    $dataTable.Columns.Add((New-Object        System.Data.DataColumn
"StationName"))

      foreach ($Row in $ds.Tables[0].Rows)
```

```
        {
            # add a new line on the object that will be exported to
excel
            New-line  $($Row[0])  $($Row[1])  $($Row[2])$($Row[3])
$($Row[4]) $($Row[5]) $($Row[6]) $($Row[7]) $($Row[8]) $($Row[9])
$($Row[10]) $($Row[11])
        }
        # export all the lines on dataTable to excel
$dataTable | Select PatientID, PatientName, PatientDOB,
AccessionNumber, StudyId, StudyDescription, BodyPart, StudyDate,
NumberofStudyRelatedInstances, StudyInstanceUid, Modality,
StationName | Export-Excel $xlfile -AutoSize -WorksheetName
$WorksheetName -TableName $TableName -Append -Numberformat "Text"
    }
    # clears all the variables (garbage collector)
    [System.GC]::Collect()
```

```
######### check if psqlODBC and importexcel is installed #########

$software = "psqlODBC_x64";
$installed = (Get-ItemProperty
HKLM:\Software\Microsoft\Windows\CurrentVersion\Uninstall\* | Where {
$_.DisplayName -eq $software }) -ne $null

If(-Not $installed) {
    Write-Error "'$software' is NOT installed. Please install it
before continue."
    exit
}
if (-Not (Get-Module -ListAvailable -Name "importexcel")) {
    Write-host "Module import-excel does not exist. Please install it
before continue."
    exit
}
#################################################################

$total_queries=19
$i=1

################# Studies with empty patient id #################

    Write-Progress -Activity "Running Queries" -status "Processing
query `"Studies with empty patient id`"" -percentComplete
($i/$total_queries*100) -CurrentOperation "$i of $total_queries"
    $query= "SELECT
        `"PatientID`" as `"PatientID`",
        `"PatientName`" as `"PatientName`",
        `"PatientBirthDate`" as `"PatientBirthDate`",
        `"AccessionNumber`" as `"AccessionNumber`",
        `"StudyId`" as `"StudyId`",
        coalesce(`"StudyDescription`", '') as `"StudyDescription`",
        coalesce(`"BodyPart`", '') as `"BodyPart`",
        coalesce(`"StudyDate`", '') as `"StudyDate`",
        `"NumberofStudyRelatedInstances`",
        `"StudyInstanceUid`",
        coalesce(`"Modality`", '') as `"Modality`",
        coalesce(`"StationName`", '') as `"StationName`"
    FROM `"ExamstoMove`" WHERE coalesce(`"PatientID`", '') = '';"
    $i++
    LIST_QUERY $query "ExamsWithEmptyPatientID"
"ExamsWithEmptyPatientID"

#################################################################
```

```
################# Studies with empty patient name ################
    Write-Progress -Activity "Running Queries" -status "Processing
query  `"Studies  with  empty  patient  name`""  -percentComplete
($i/$total_queries*100) -CurrentOperation "$i of $total_queries"
    $query= "SELECT
        `"PatientID`" as `"PatientID`",
        `"PatientName`" as `"PatientName`",
        `"PatientBirthDate`" as `"PatientBirthDate`",
        `"AccessionNumber`" as `"AccessionNumber`",
        `"StudyId`" as `"StudyId`",
        coalesce(`"StudyDescription`", '') as `"StudyDescription`",
        coalesce(`"BodyPart`", '') as `"BodyPart`",
        coalesce(`"StudyDate`", '') as `"StudyDate`",
        `"NumberofStudyRelatedInstances`",
        `"StudyInstanceUid`",
        coalesce(`"Modality`", '') as `"Modality`",
        coalesce(`"StationName`", '') as `"StationName`"
    FROM `"ExamstoMove`" WHERE coalesce(`"PatientName`", '') = '';"
    $i++
    LIST_QUERY          $query          "ExamsWithEmptyPatientName"
"ExamsWithEmptyPatientName"

##################################################################

############# Studies with empty patient birth date ##############
    Write-Progress -Activity "Running Queries" -status "Processing
query `"Studies with empty patient birth date`"" -percentComplete
($i/$total_queries*100) -CurrentOperation "$i of $total_queries"
    $query= "SELECT
        `"PatientID`" as `"PatientID`",
        `"PatientName`" as `"PatientName`",
        `"PatientBirthDate`" as `"PatientBirthDate`",
        `"AccessionNumber`" as `"AccessionNumber`",
        `"StudyId`" as `"StudyId`",
        coalesce(`"StudyDescription`", '') as `"StudyDescription`",
        coalesce(`"BodyPart`", '') as `"BodyPart`",
        coalesce(`"StudyDate`", '') as `"StudyDate`",
        `"NumberofStudyRelatedInstances`",
        `"StudyInstanceUid`",
        coalesce(`"Modality`", '') as `"Modality`",
        coalesce(`"StationName`", '') as `"StationName`"
    FROM  `"ExamstoMove`"  WHERE  coalesce(`"PatientBirthDate`",  '')  =
'';"
    $i++
    LIST_QUERY        $query        "ExamsWithEmptyPatientBirthDate"
"ExamsWithEmptyPatientBirthDate"

##################################################################
```

```
############### Studies with empty patient gender ################
    Write-Progress -Activity "Running Queries" -status "Processing
query `"Studies with `"Studies with empty patient gender`"" -
percentComplete ($i/$total_queries*100) -CurrentOperation "$i of
$total_queries"
    $query= "SELECT
        `"PatientID`" as `"PatientID`",
        `"PatientName`" as `"PatientName`",
        `"PatientBirthDate`" as `"PatientBirthDate`",
        `"AccessionNumber`" as `"AccessionNumber`",
        `"StudyId`" as `"StudyId`",
        coalesce(`"StudyDescription`", '') as `"StudyDescription`",
        coalesce(`"BodyPart`", '') as `"BodyPart`",
        coalesce(`"StudyDate`", '') as `"StudyDate`",
        `"NumberofStudyRelatedInstances`",
        `"StudyInstanceUid`",
        coalesce(`"Modality`", '') as `"Modality`",
        coalesce(`"StationName`", '') as `"StationName`"
    FROM `"ExamstoMove`" WHERE coalesce(`"PatientGender`", '') = '';"
    $i++
    LIST_QUERY              $query              "ExamsWithEmptyPatientSex"
"ExamsWithEmptyPatientSex"

#################################################################

############## Studies with empty accession number ###############
    Write-Progress -Activity "Running Queries" -status "Processing
query `"Studies with `"Studies with empty accession number`"" -
percentComplete ($i/$total_queries*100) -CurrentOperation "$i of
$total_queries"
    $query= "SELECT
        `"PatientID`" as `"PatientID`",
        `"PatientName`" as `"PatientName`",
        `"PatientBirthDate`" as `"PatientBirthDate`",
        `"AccessionNumber`" as `"AccessionNumber`",
        `"StudyId`" as `"StudyId`",
        coalesce(`"StudyDescription`", '') as `"StudyDescription`",
        coalesce(`"BodyPart`", '') as `"BodyPart`",
        coalesce(`"StudyDate`", '') as `"StudyDate`",
        `"NumberofStudyRelatedInstances`",
        `"StudyInstanceUid`",
        coalesce(`"Modality`", '') as `"Modality`",
        coalesce(`"StationName`", '') as `"StationName`"
    FROM `"ExamstoMove`" WHERE coalesce(`"AccessionNumber`", '') =
'';"
    $i++
    LIST_QUERY              $query              "StudiesWithEmptytAccNo"
"StudiesWithEmptytAccNo"

#################################################################
```

```
################## Studies with empty modality ##################
    Write-Progress -Activity "Running Queries" -status "Processing
query `"Studies with empty modality`"" -percentComplete
($i/$total_queries*100) -CurrentOperation "$i of $total_queries"
    $query= "SELECT
        `"PatientID`" as `"PatientID`",
        `"PatientName`" as `"PatientName`",
        `"PatientBirthDate`" as `"PatientBirthDate`",
        `"AccessionNumber`" as `"AccessionNumber`",
        `"StudyId`" as `"StudyId`",
        coalesce(`"StudyDescription`", '') as `"StudyDescription`",
        coalesce(`"BodyPart`", '') as `"BodyPart`",
        coalesce(`"StudyDate`", '') as `"StudyDate`",
        `"NumberofStudyRelatedInstances`",
        `"StudyInstanceUid`",
        coalesce(`"Modality`", '') as `"Modality`",
        coalesce(`"StationName`", '') as `"StationName`"
    FROM `"ExamstoMove`" WHERE coalesce(`"Modality`", '') = '';"
    $i++
    LIST_QUERY $query "StudiesWithEmptyModality"
"StudiesWithEmptyModality"

#################################################################

############## Studies with zero related instances ###############
    Write-Progress -Activity "Running Queries" -status "Processing
query `"Studies with zero related instances`"" -percentComplete
($i/$total_queries*100) -CurrentOperation "$i of $total_queries"
    $query= "SELECT
        `"PatientID`" as `"PatientID`",
        `"PatientName`" as `"PatientName`",
        `"PatientBirthDate`" as `"PatientBirthDate`",
        `"AccessionNumber`" as `"AccessionNumber`",
        `"StudyId`" as `"StudyId`",
        coalesce(`"StudyDescription`", '') as `"StudyDescription`",
        coalesce(`"BodyPart`", '') as `"BodyPart`",
        coalesce(`"StudyDate`", '') as `"StudyDate`",
        `"NumberofStudyRelatedInstances`",
        `"StudyInstanceUid`",
        coalesce(`"Modality`", '') as `"Modality`",
        coalesce(`"StationName`", '') as `"StationName`"
    FROM `"ExamstoMove`" WHERE `"NumberofStudyRelatedInstances`" =
0;"
    $i++
    LIST_QUERY $query "ExamsWithoutInstances" "ExamsWithoutInstances"

#################################################################
```

```
########### Studies with patient name exceeding 64 chars #########
    Write-Progress -Activity "Running Queries" -status "Processing query `"Studies with patient name exceeding 64 chars`"" -percentComplete ($i/$total_queries*100) -CurrentOperation "$i of $total_queries"
    $query= "SELECT
        `"PatientID`" as `"PatientID`",
        `"PatientName`" as `"PatientName`",
        `"PatientBirthDate`" as `"PatientBirthDate`",
        `"AccessionNumber`" as `"AccessionNumber`",
        `"StudyId`" as `"StudyId`",
        coalesce(`"StudyDescription`", '') as `"StudyDescription`",
        coalesce(`"BodyPart`", '') as `"BodyPart`",
        coalesce(`"StudyDate`", '') as `"StudyDate`",
        `"NumberofStudyRelatedInstances`",
        `"StudyInstanceUid`",
        coalesce(`"Modality`", '') as `"Modality`",
        coalesce(`"StationName`", '') as `"StationName`"
    FROM `"ExamstoMove`" WHERE length(`"PatientName`")>64;"
    $i++
    LIST_QUERY $query "NonConformantPatientName" "NonConformantPatientName"

##################################################################

######## Studies with accession number exceeding 16 chars #######

 Write-Progress -Activity "Running Queries" -status "Processing query `"Studies with accession number exceeding 16 chars`"" -percentComplete ($i/$total_queries*100) -CurrentOperation "$i of $total_queries"
    $query= "SELECT
        `"PatientID`" as `"PatientID`",
        `"PatientName`" as `"PatientName`",
        `"PatientBirthDate`" as `"PatientBirthDate`",
        `"AccessionNumber`" as `"AccessionNumber`",
        `"StudyId`" as `"StudyId`",
        coalesce(`"StudyDescription`", '') as `"StudyDescription`",
        coalesce(`"BodyPart`", '') as `"BodyPart`",
        coalesce(`"StudyDate`", '') as `"StudyDate`",
        `"NumberofStudyRelatedInstances`",
        coalesce(`"Modality`", '') as `"Modality`",
        coalesce(`"StationName`", '') as `"StationName`"
    FROM `"ExamstoMove`" WHERE length(`"AccessionNumber`")>16;"
    $i++
    LIST_QUERY $query "StudiesWithMoreThen16Char" "StudiesWithMoreThen16Char"

##################################################################
```

```
###### Accession numbers with multiple study instance uids #######
    Write-Progress -Activity "Running Queries" -status "Processing
query `"Accession numbers with multiple study instance uids`"" -
percentComplete ($i/$total_queries*100) -CurrentOperation "$i of
$total_queries"
    $query= "SELECT
        `"PatientID`" as `"PatientID`",
        `"PatientName`" as `"PatientName`",
        `"PatientBirthDate`" as `"PatientBirthDate`",
        a.`"AccessionNumber`" as `"AccessionNumber`",
        `"StudyId`" as `"StudyId`",
        coalesce(`"StudyDescription`", '') as `"StudyDescription`",
        coalesce(`"BodyPart`", '') as `"BodyPart`",
        coalesce(`"StudyDate`", '') as `"StudyDate`",
        `"NumberofStudyRelatedInstances`",
        `"StudyInstanceUid`",
        coalesce(`"Modality`", '') as `"Modality`",
        coalesce(`"StationName`", '') as `"StationName`"
    FROM `"ExamstoMove`" as a
    inner join (
        SELECT          `"AccessionNumber`",          count(distinct
`"StudyInstanceUid`") as `"Count`" FROM `"ExamstoMove`"
            where coalesce(`"AccessionNumber`", '')<>''
            And `"StudyDate`">'20171231'
            GROUP BY `"AccessionNumber`"
            having count(distinct `"StudyInstanceUid`")>1
    ) as b on a.`"AccessionNumber`" = b.`"AccessionNumber`""
    $i++
    LIST_QUERY          $query          "StudiesWithMultiStudyIUIDs"
"StudiesWithMultiStudyIUIDs"

##################################################################

####### Study instance uids with multiple accession numbers ######

    Write-Progress -Activity "Running Queries" -status "Processing
query `"Study instance uids with multiple accession numbers`"" -
percentComplete ($i/$total_queries*100) -CurrentOperation "$i of
$total_queries"
    $query= "SELECT
        `"PatientID`" as `"PatientID`",
        `"PatientName`" as `"PatientName`",
        `"PatientBirthDate`" as `"PatientBirthDate`",
        `"AccessionNumber`" as `"AccessionNumber`",
        `"StudyId`" as `"StudyId`",
        coalesce(`"StudyDescription`", '') as `"StudyDescription`",
        coalesce(`"BodyPart`", '') as `"BodyPart`",
        coalesce(`"StudyDate`", '') as `"StudyDate`",
        `"NumberofStudyRelatedInstances`",
        coalesce(`"Modality`", '') as `"Modality`",
        a.`"StudyInstanceUid`",
        coalesce(`"StationName`", '') as `"StationName`"
    FROM `"ExamstoMove`" as a
    inner join (
```

```
        SELECT          `"StudyInstanceUid`",         count(distinct
`"AccessionNumber`") as `"Count`" FROM `"ExamstoMove`"
            where `"StudyDate`">'20171231'
            GROUP BY `"StudyInstanceUid`"
            having count(distinct `"AccessionNumber`")>1
    ) as b on b.`"StudyInstanceUid`"=a.`"StudyInstanceUid`";"
    $i++
    LIST_QUERY            $query            "StudiesWithEmptyStudyId"
"StudiesWithEmptyStudyId"

##################################################################

################ Studies older than cutoff date ##################
    Write-Progress -Activity "Running Queries" -status "Processing
query   `"Studies   older   than   cutoff   date`""   -percentComplete
($i/$total_queries*100) -CurrentOperation "$i of $total_queries"
    $query= "SELECT
        `"PatientID`" as `"PatientID`",
        `"PatientName`" as `"PatientName`",
        `"PatientBirthDate`" as `"PatientBirthDate`",
        `"AccessionNumber`" as `"AccessionNumber`",
        `"StudyId`" as `"StudyId`",
        coalesce(`"StudyDescription`", '') as `"StudyDescription`",
        coalesce(`"BodyPart`", '') as `"BodyPart`",
        coalesce(`"StudyDate`", '') as `"StudyDate`",
        `"NumberofStudyRelatedInstances`",
        `"StudyInstanceUid`",
        coalesce(`"Modality`", '') as `"Modality`",
        coalesce(`"StationName`", '') as `"StationName`"
    FROM `"ExamstoMove`" WHERE `"StudyDate`" < '"+ $cutofdate + "';"
    $i++
    LIST_QUERY            $query            "StudiesOlderthanCutoff"
"StudiesOlderthanCutoff"

##################################################################
```

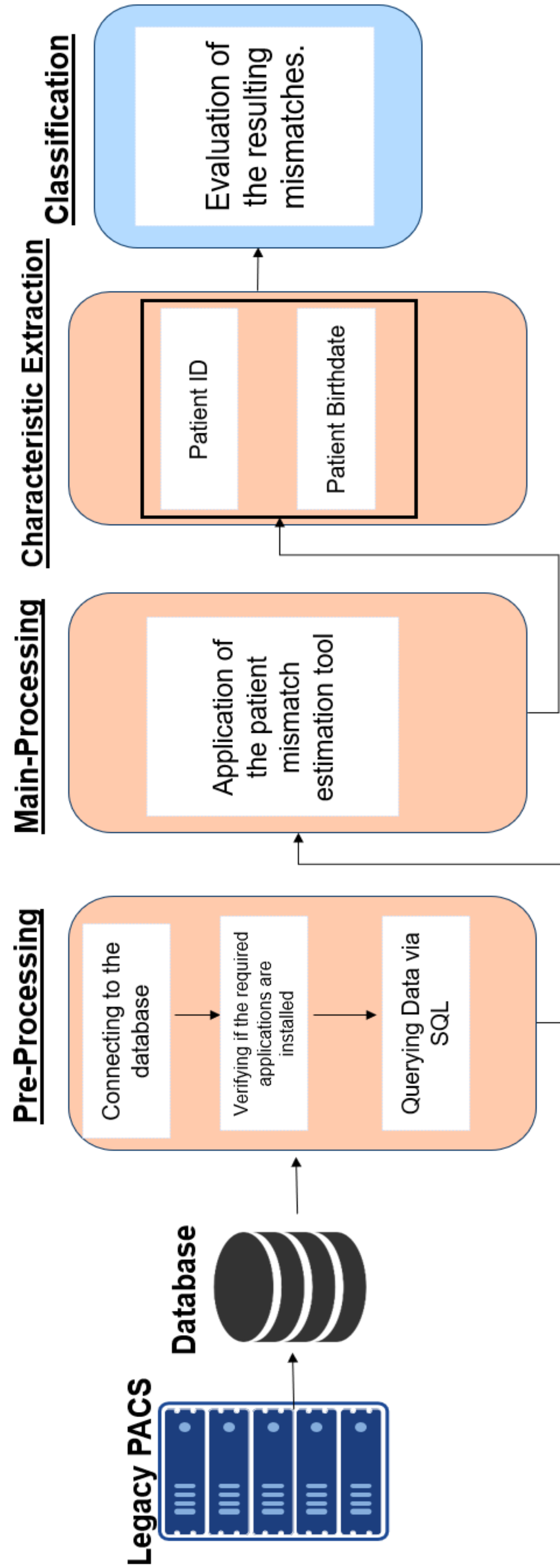
```
######### Studies with non-conformant study instance uid #########

 Write-Progress -Activity "Running Queries" -status "Processing query
`"Studies with non-conformant study instance uid`"" -percentComplete
($i/$total_queries*100) -CurrentOperation "$i of $total_queries"

 $query= "SELECT
        `"PatientID`" as `"PatientID`",
        `"PatientName`" as `"PatientName`",
        `"PatientBirthDate`" as `"PatientBirthDate`",
        `"AccessionNumber`" as `"AccessionNumber`",
        `"StudyId`" as `"StudyId`",
        coalesce(`"StudyDescription`", '') as `"StudyDescription`",
        coalesce(`"BodyPart`", '') as `"BodyPart`",
        coalesce(`"StudyDate`", '') as `"StudyDate`",
        `"NumberofStudyRelatedInstances`",
        `"StudyInstanceUid`",
        coalesce(`"Modality`", '') as `"Modality`",
        coalesce(`"StationName`", '') as `"StationName`"
    FROM `"ExamstoMove`" WHERE `"StudyInstanceUid`"  similar to'%[^0-
9.]%';"
    $i++
    LIST_QUERY $query "NonConformantSIUID" "NonConformantSIUID"

##################################################################

########## Studies with non-conformant accession number ##########
    Write-Progress -Activity "Running Queries" -status "Processing
query `"Studies with non-conformant accession number`"" -
percentComplete ($i/$total_queries*100) -CurrentOperation "$i of
$total_queries"
    $query= "SELECT
        `"PatientID`" as `"PatientID`",
        `"PatientName`" as `"PatientName`",
        `"PatientBirthDate`" as `"PatientBirthDate`",
        `"AccessionNumber`" as `"AccessionNumber`",
        `"StudyId`" as `"StudyId`",
        coalesce(`"StudyDescription`", '') as `"StudyDescription`",
        coalesce(`"BodyPart`", '') as `"BodyPart`",
        coalesce(`"StudyDate`", '') as `"StudyDate`",
        `"NumberofStudyRelatedInstances`",
        `"StudyInstanceUid`",
        coalesce(`"Modality`", '') as `"Modality`",
        coalesce(`"StationName`", '') as `"StationName`"
    FROM `"ExamstoMove`"
    WHERE
        coalesce(`"AccessionNumber`", '') <> '' and
        `"AccessionNumber`" not similar to '" + $accno_pattern + "';"
    $i++
    LIST_QUERY $query "NonConformantAccNo" "NonConformantAccNo"

##################################################################


############# Studies with non-conformant patient id #############
```

```
    Write-Progress -Activity "Running Queries" -status "Processing
query `"Studies with non-conformant patient id`"" -percentComplete
($i/$total_queries*100) -CurrentOperation "$i of $total_queries"
    $query= "SELECT
        `"PatientID`" as `"PatientID`",
        `"PatientName`" as `"PatientName`",
        `"PatientBirthDate`" as `"PatientBirthDate`",
        `"AccessionNumber`" as `"AccessionNumber`",
        `"StudyId`" as `"StudyId`",
        coalesce(`"StudyDescription`", '') as `"StudyDescription`",
        coalesce(`"BodyPart`", '') as `"BodyPart`",
        coalesce(`"StudyDate`", '') as `"StudyDate`",
        `"NumberofStudyRelatedInstances`",
        `"StudyInstanceUid`",
        coalesce(`"Modality`", '') as `"Modality`",
        coalesce(`"StationName`", '') as `"StationName`"
    FROM `"ExamstoMove`"
    WHERE
        coalesce(`"PatientID`", '') <> '' AND
        `"PatientID`" not SIMILAR TO '"+ $pid_pattern +"';"
    $i++
    LIST_QUERY             $query             "NonConformantPatientId"
"NonConformantPatientId"

###################################################################
####### Studies with non-conformant or strange patient name ######
    Write-Progress -Activity "Running Queries" -status "Processing
query `"Studies with non-conformant or strange patient name`"" -
percentComplete ($i/$total_queries*100) -CurrentOperation "$i of
$total_queries"
    $query= "SELECT
        `"PatientID`" as `"PatientID`",
        `"PatientName`" as `"PatientName`",
        `"PatientBirthDate`" as `"PatientBirthDate`",
        `"AccessionNumber`" as `"AccessionNumber`",
        `"StudyId`" as `"StudyId`",
        coalesce(`"StudyDescription`", '') as `"StudyDescription`",
        coalesce(`"BodyPart`", '') as `"BodyPart`",
        coalesce(`"StudyDate`", '') as `"StudyDate`",
        `"NumberofStudyRelatedInstances`",
        `"StudyInstanceUid`",
        coalesce(`"Modality`", '') as `"Modality`",
        coalesce(`"StationName`", '') as `"StationName`"
    FROM `"ExamstoMove`"
    WHERE
        lower(`"PatientName`") similar to '%test%' OR
        lower(`"PatientName`") similar to '%unknown%' OR
        lower(`"PatientName`") similar to '%synapse%' OR
        lower(`"PatientName`") similar to '%fuji%' OR
        lower(`"PatientName`") similar to '%sectra%' OR
        lower(`"PatientName`") similar to '%siemens%' OR
        lower(`"PatientName`") similar to '%philips%' OR
        lower(`"PatientName`") similar to '%service%' OR
        lower(`"PatientName`") similar to '%agfa%';"
    $i++
```

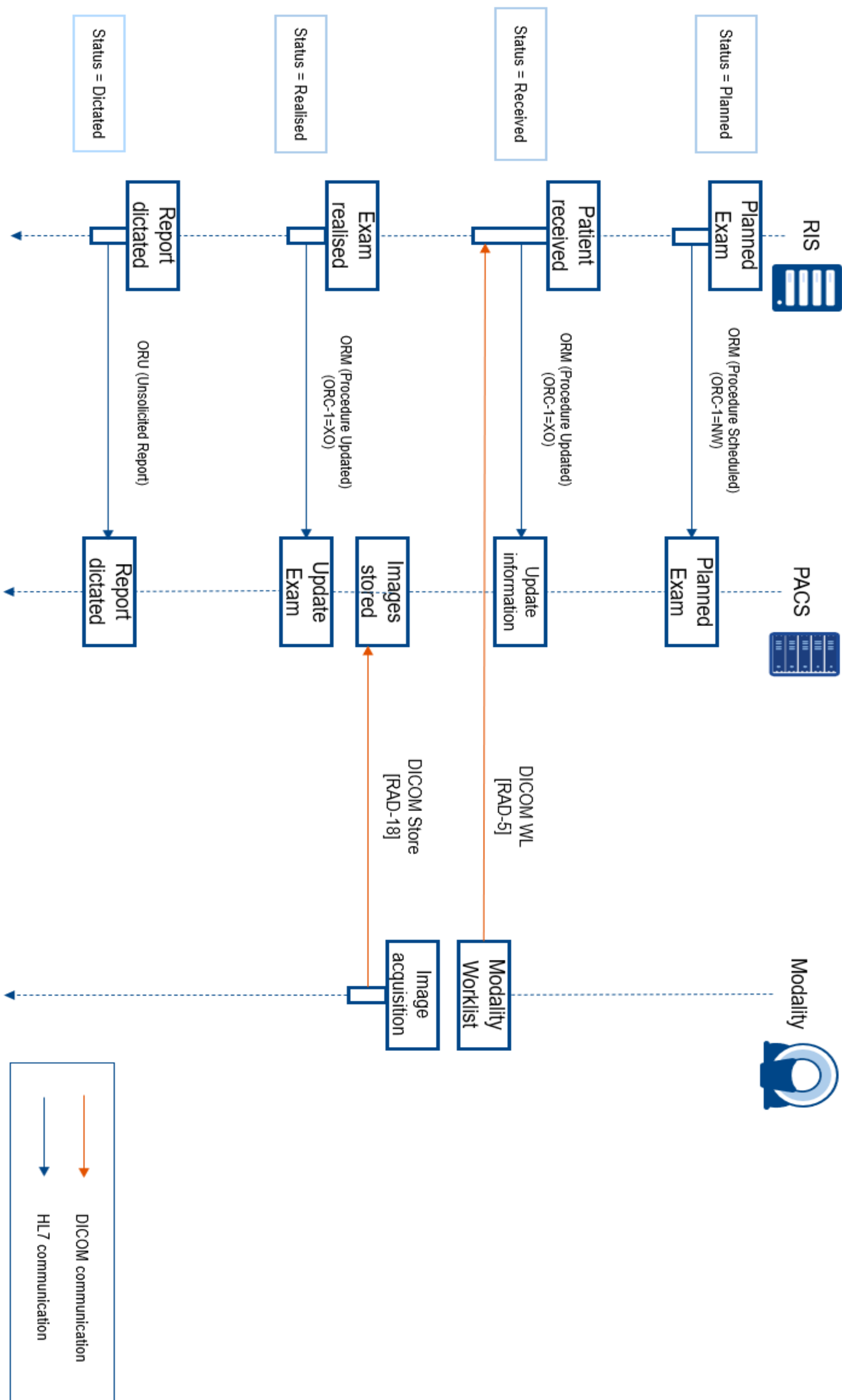
```
    LIST_QUERY           $query          "NonConformantPatientName"
"NonConformantPatientName"

##################################################################

########### Studies with non-conformant patient gender ###########
    Write-Progress -Activity "Running Queries" -status "Processing
query `"Studies with non-conformant patient gender`"" -
percentComplete ($i/$total_queries*100) -CurrentOperation "$i of
$total_queries"
    $query= "SELECT
        `"PatientID`" as `"PatientID`",
        `"PatientName`" as `"PatientName`",
        `"PatientBirthDate`" as `"PatientBirthDate`",
        `"AccessionNumber`" as `"AccessionNumber`",
        `"StudyId`" as `"StudyId`",
        coalesce(`"StudyDescription`", '') as `"StudyDescription`",
        coalesce(`"BodyPart`", '') as `"BodyPart`",
        coalesce(`"StudyDate`", '') as `"StudyDate`",
        `"NumberofStudyRelatedInstances`",
        `"StudyInstanceUid`",
        coalesce(`"Modality`", '') as `"Modality`",
        coalesce(`"StationName`", '') as `"StationName`"
    FROM `"ExamstoMove`"
    WHERE
        coalesce(`"PatientGender`", '') <> '' and
        `"PatientGender`" not similar to '[FMOU]';"
    $i++
    LIST_QUERY           $query          "NonConformantPatientSex"
"NonConformantPatientSex"

##################################################################

########### Studies with mismatch on Patient BirthDate ###########
    Write-Progress -Activity "Running Queries" -status "Processing
query `"Studies with mismatch on Patient BirthDate`"" -percentComplete
($i/$total_queries*100) -CurrentOperation "$i of $total_queries"
    $query= "SELECT
        `"PatientID`" as `"PatientID`",
        `"PatientName`" as `"PatientName`",
        `"PatientBirthDate`" as `"PatientBirthDate`",
        `"AccessionNumber`" as `"AccessionNumber`",
        `"StudyId`" as `"StudyId`",
        coalesce(`"StudyDescription`", '') as `"StudyDescription`",
        coalesce(`"BodyPart`", '') as `"BodyPart`",
        coalesce(`"StudyDate`", '') as `"StudyDate`",
        `"NumberofStudyRelatedInstances`",
        `"StudyInstanceUid`",
        coalesce(`"Modality`", '') as `"Modality`",
        coalesce(`"StationName`", '') as `"StationName`"
        from `"ExamstoMove`"
        where `"PatientID`" in
        (
        SELECT `"PatientID`" FROM `"ExamstoMove`"
            GROUP BY `"PatientID`"
```

```
            having count(distinct `"PatientBirthDate`")>1);"
    $i++
    LIST_QUERY $query "DOB_Mismatch" "DOB_Mismatch"

####################################################################

########### Studies with mismatch on MRN ###########

Write-Progress -Activity "Running Queries" -status "Processing query
`"Studies    with    mismatch    on    MRN`""    -percentComplete
($i/$total_queries*100) -CurrentOperation "$i of $total_queries"
    $query= "SELECT
        `"PatientID`" as `"PatientID`",
        `"PatientName`" as `"PatientName`",
        `"PatientBirthDate`" as `"PatientBirthDate`",
        `"AccessionNumber`" as `"AccessionNumber`",
        `"StudyId`" as `"StudyId`",
        coalesce(`"StudyDescription`", '') as `"StudyDescription`",
        coalesce(`"BodyPart`", '') as `"BodyPart`",
        coalesce(`"StudyDate`", '') as `"StudyDate`",
        `"NumberofStudyRelatedInstances`",
        `"StudyInstanceUid`",
        coalesce(`"Modality`", '') as `"Modality`",
        coalesce(`"StationName`",   '')   as   `"StationName`"   from
`"ExamstoMove`"
        where coalesce(`"AccessionNumber`",'')<>''
        and `"AccessionNumber`" in
        (
            select `"AccessionNumber`" from `"ExamstoMove`"
        where coalesce(`"AccessionNumber`",'')<>''
                    GROUP BY `"AccessionNumber`"
        having   count(`"AccessionNumber`")>1   and   count(distinct
`"PatientID`")>1
        )
        order by 2;"
    $i++
    LIST_QUERY $query "MRN_Mismatch" "MRN_Mismatch"
```

A.3 Flowchart of the Mismatch Estimation Tool



A.4 Basic image acquisition workflow according to IHE



References

Adelman, J. S., Applebaum, J. R., Southern, W. N., Schechter, C. B., Aschner, J. L., Berger, M. A., Racine, A. D., Chacko, B., Dadlez, N. M., Goffman, D., Babineau, J., Green, R. A., Vawdrey, D. K., Manzano, W., Barchi, D., Albanese, C., Bates, D. W., & Salmasian, H. (2019). Risk of Wrong-Patient Orders Among Multiple vs Singleton Births in the Neonatal Intensive Care Units of 2 Integrated Health Care Systems. *JAMA Pediatrics.* Advance online publication. https://doi.org/10.1001/jamapediatrics.2019.2733

Benson, T. (2012). *Principles of Health Interoperability HL7 and SNOMED*. Springer London. https://doi.org/10.1007/978-1-4471-2801-4

Berkowitz, S. J., Wei, J. L., & Halabi, S. (2018). Migrating to the Modern PACS: Challenges and Opportunities. *Radiographics : A Review Publication of the Radiological Society of North America, Inc*, *38*(6), 1761–1772. https://doi.org/10.1148/rg.2018180161

Branstetter, B. F. (2010). *Practical Imaging Informatics*. Springer New York. https://doi.org/10.1007/978-1-4419-0485-0

Cambridge Dictionary. (2021). *Definition of workflow.* Cambridge Dictionary. https://dictionary.cambridge.org/dictionary/english/workflow

Choplin, R. H., Boehme, J. M., & Maynard, C. D. (1992). Picture archiving and communication systems: An overview. *Radiographics : A Review Publication of the Radiological Society of North America, Inc*, *12*(1), 127–129. https://doi.org/10.1148/radiographics.12.1.1734458

Cloudflare, I. (2020). *What is the OSI Model ?* https://www.cloudflare.com/en-gb/learning/ddos/glossary/open-systems-interconnection-model-osi/ [Date Visited: 06/20/2021]

Darthmouth College. (2020). *DICOM Resources: DICOM Attributes.* Darthmouth College. https://northstar-www.dartmouth.edu/doc/idl/html_6.2/DICOM_Attributes.html [Date Visited: 06/14/2021]

Dicken, A. J., Evans, J. P. O., Rogers, K. D., Prokopiou, D., Godber, S. X., & Wilson, M. (2017). Depth resolved snapshot energy-dispersive X-ray diffraction using a conical shell beam. *Optics Express*, *25*(18), 21321–21328. https://doi.org/10.1364/OE.25.021321

DICOM Standards Committee. (2020). *About DICOM*. https://www.dicomstandard.org/ [Date Visited: 06/18/2021]

DICOM Standards Committee. (2020). *DICOM PS3.1 2020d - Introduction and Overview*. [Date Visited: 06/18/2021]

Dixon, B. (2016). *Health information exchange - navigating and managing a network of health i* (B. (. P. R. M. F. S. Dixon, Ed.). Elsevier Science Publishing Co.

Dziedzic, A., Elmore, A. J., & Stonebraker, M. (2016, September). Data transformation and migration in polystores. In *2016 IEEE High Performance Extreme Computing Conference (HPEC)* (pp. 1–6). IEEE. https://doi.org/10.1109/HPEC.2016.7761594

Erin Benson. (2017). *Mismatched: How Patient Identi¦cation Errors Are Costing Patients And Health Systems.* Healthitoutcomes. https://www.healthitoutcomes.com/doc/mismatched-how-patient-identification-errors-are-costing-patients-and-health-systems-0001

Errata (2002). *Journal of the American Medical Informatics Association*, *9*(3), 307. https://doi.org/10.1136/jamia.2002.0090307

Health Level Seven International. (2020). *About HL7*. http://www.hl7.org/about/index.cfm?ref=common [Date Visited: 06/20/2021]

Hillestad, R., Bigelow, J. H., Chaudhry, B., Dreyer, P., Greenberg, M. D., Meili, R. C., Ridgely, S. M., Rothenberg, J., & Taylor, R. (2008). *Identity Crisis: An Examination of the Costs and Benefits of a Unique Patient Identifier for the U.S. Health Care System.* RAND Corporation. https://www.rand.org/pubs/monographs/MG753.html [Date Visited: 06/18/2021]

Huang, H. K. (2004). *PACS and imaging informatics: Basic principles and applications* (2nd ed.). Wiley-Liss.

Huang, H. K. (2010). *PACS and Imaging Informatics: Basic Principles and Applications, 2nd Editio* (2. uppl). John Wiley & Sons.

Integrating the Healthcare Enterprise IHE Radiology (RAD) Technical Framework, September 18, 2020.

Imkamp, D., Berthold, J., Heizmann, M., Kniel, K., Manske, E., Peterek, M., Schmitt, R., Seidler, J., & Sommer, K.-D. (2016). Challenges and trends in manufacturing measurement technology – the “Industrie 4.0” concept. *Journal of Sensors and Sensor Systems*, *5*(2), 325–335. https://doi.org/10.5194/jsss-5-325-2016

(2014).

Integrating the Healthcare Enterprise. (2021). *Integrating the Healthcare Enterprise*. https://www.ihe.net/ [Date Visited: 03/04/2021]

Jackie Leckas. (2021). *Preventing Patient Data Mismatches When Deploying Image Exchange Enterprise Wide.* Life Image. https://www.lifeimage.com/2016/06/08/preventing-patient-data-mismatches-when-deploying-image-exchange-enterprise-wide

Just, B. H., Marc, D., Munns, M., & Sandefer, R. (2016). Why Patient Matching Is a Challenge: Research on Master Patient Index (MPI) Data Discrepancies in Key Identifying Fields. *Perspectives in Health Information Management*, *13*(Spring).

Kim, Y.-A. (2005). Data migration to minimize the total completion time. *Journal of Algorithms*, *55*(1), 42–57. https://doi.org/10.1016/j.jalgor.2004.07.009

Kiourtis, A., Mavrogiorgou, A., Menychtas, A., Maglogiannis, I., & Kyriazis, D. (2019). Structurally Mapping Healthcare Data to HL7 FHIR through Ontology Alignment. *Journal of Medical Systems*, *43*(3), 62. https://doi.org/10.1007/s10916-019-1183-y

Krishna, C. R., Dutta, M., & Kumar, R. (Eds.). (2019). *Lecture Notes in Networks and Systems. Proceedings of 2nd International Conference on Communication, Computing and Networking*. Springer Singapore. https://doi.org/10.1007/978-981-13-1217-5

Lippi, G., Mattiuzzi, C., Bovo, C., & Favaloro, E. J. (2017). Managing the patient identification crisis in healthcare and laboratory medicine. *Clinical Biochemistry*, *50*(10-11), 562–567. https://doi.org/10.1016/j.clinbiochem.2017.02.004

Martins, A. C. (2014). *Novos sistemas de arquivo e comunicação de imagens médicas – uma abrangencia cada vez maior.*

Mazzoncini de Azevedo-Marques, P., & Covas Salomão, S. (2009). PACS: Sistemas de Arquivamento e Distribuição de Imagens: PACS: Picture Archiving and Communication Systems. *Revista Brasileira De Física Médic*, pp. 131–139.

Microsoft. (2021). *What is Powershell?* https://docs.microsoft.com/en-us/powershell/scripting/overview?view=powershell-7.1 [Date Visited: 06/19/2021]

Morris, J. (2012). *Practical data migration* (2nd ed.). BCS Learning and Development.

Morrison, A. P., Tanasijevic, M. J., Goonan, E. M., Lobo, M. M., Bates, M. M., Lipsitz, S. R., Bates, D. W., & Melanson, S. E. F. (2010). Reduction in specimen labeling errors after implementation of a positive patient identification system in phlebotomy. *American Journal of Clinical Pathology*, *133*(6), 870–877. https://doi.org/10.1309/AJCPC95YYMSLLRCX

National Commitee On Vital And Health Statistiacs (2000, July 6). Uniform Data Standards for Patient Medical Record Information. https://ncvhs.hhs.gov/wp-content/uploads/2014/05/hipaa000706.pdf [Date Visited: 06/18/2021]

National Electrical Manufacturers Association. (2020a). *DICOM standard*. https://www.dicomstandard.org/current/ [Date Visited: 06/18/2021]

National Electrical Manufacturers Association. (2020b). *What is DICOM?* http://dicom.nema.org/dicom/geninfo/brochure/BROCH96.HTM [Date Visited: 06/21/2021]

Oliveira, R., Ferreira, D., Ferreira, R., & Cruz-Correia, R. (2016, June). Open-Source Based Integration Solution for Hospitals. In *2016 IEEE 29th International Symposium on Computer-Based Medical Systems (CBMS)* (pp. 294–299). IEEE. https://doi.org/10.1109/CBMS.2016.44

Pianykh, O. S. (2012). Digital Imaging and Communications in Medicine (DICOM). Advance online publication. https://doi.org/10.1007/978-3-642-10850-1

Practical Guide to Clinical Computing Systems. (2015). Elsevier.

(2013, June - 2013, June). *Proceedings of the 26th IEEE International Symposium on Computer-Based Medical Systems*. IEEE.

R. da C. Oliveira. (2016). *Transformation of clinical data from JL7 messages to openEHR compositions.*

Redbooks, I. (2015). *Ds8870 data migration techniques*. Vervante.

Schreiber, R., & Garber, L. (2020). Data Migration: A Thorny Issue in Electronic Health Record Transitions—Case Studies and Review of the Literature. *ACI Open*, *04*(01), e48-e58. https://doi.org/10.1055/s-0040-1710007

Sectra (September 2019). System Administrator's Guide.

Sectra. (2021). *Sectra PACS: Best in KLAS*. https://medical.sectra.com/about-sectra/sectra-pacs-best-in-klas/

Shafarman, M., Tucker, M., Reis, L., & Henderson, M. (2000). *HL7 Messaging Standard Version 2.5 Chapter 2. Control,*

Softneta, m. (2020). *DICOM Library* [SOPs]. https://www.dicomlibrary.com/dicom/sop [Date Visited: 05/04/2021]

Taieb-Maimon, M., Plaisant, C., Hettinger, A. Z., & Shneiderman, B. (2018). Increasing Recognition of Wrong-Patient Errors through Improved Interface Design of a Computerized Provider Order Entry System. *International Journal of Human–Computer Interaction*, *34*(5), 383–398. https://doi.org/10.1080/10447318.2017.1349249

Techopedia. (2017). *Definition - Data Migration.* Techopedia. https://www.techopedia.com/definition/1180/data-migration [Date Visited: 06/18/2021]

Teli, P., Thomas, M. V., & Chandrasekaran, K. (2016). Big Data Migration between Data Centers in Online Cloud Environment. *Procedia Technology*, *24*, 1558–1565. https://doi.org/10.1016/j.protcy.2016.05.135

Thalheim, B., & Wang, Q. (2013). Data migration: A theoretical perspective. *Data & Knowledge Engineering*, *87*, 260–278. https://doi.org/10.1016/j.datak.2012.12.003