RISPad – Radiology Information Storage Notepad

A DICOM-Compatible, Web-Technology Based Documentation Tool for the Integration of Clinical and Medico-Legal Evidence Documents into an Associated Radiological Examination

Master Thesis

For attainment of the academic degree of
Master of Science in Engineering (MSc)

in the Master Program Digital Healthcare
at St. Pölten University of Applied Sciences

by

Univ.Prof. Dr. Christian Nasel, PhD
dh151815

First advisor: Dr. Helmut Ritschl, MA MSc

St.Pölten, 14.05.2017

Declaration

I declare that I have developed and written the enclosed Master Thesis completely by myself, and have not used sources or means without declaration in the text. Any thoughts from others or literal quotations are clearly marked. This work was not used in the same or in a similar version to achieve an academic grading or is being published elsewhere.

.. Vienna, 14.05.2017....... ....

Place, Date Signature

Abstract

Radiology lacks a coherent system for the documentation of relevant clinical and scientific information encountered during an examination or intervention. This can be achieved by a simple digital health care (DHC) software tool capturing images form clinical evidence documents and biological material extracted from the human body during an intervention. The collection of this data can potentially gain importance in medico-legal law-suits and is also supportive in radiological science. Therefore, the feasibility to implement such a tool was investigated in order get a positive proof of concept (PoC) according to standards of the Integrating the Health Enterprise (IHE) consortium.

Following the intention to create an easy-to-use, low-budget, cost-effective, light-weight application the RISPad-application was designed and tested following the IHE – proposed four step process for the implementation of a health care software.

Cost effectiveness was reached by using only approved, freely available software tools and integrated development of high reputation for the development of RisPad. Additionally, a reasonably priced high resolution document camera device was added to the system, which is recognized as a webcam by the PC. In order to avoid the installation of excessive amounts of software on every computer in the department, where RISPad was tested, a server-client solution with the client running in a standard browser window was implemented. The RISPad-Clt was equipped with a straight forward and clear structured Graphic User Interface (GUI) that widely provided those functions absolute necessary for the documentation process.

Beta-testing at a typical radiological department assigned a positive PoC to the RISPad-project. The further testing and development of the RISPad-software to a real world application is pending.

Kurzfassung

In der Radiologie fehlt ein schlüssiges System für die Dokumentation von relevanten klinischen und wissenschaftlichen Informationen, welche während einer Untersuchung oder eines Eingriffs evident werden. Eine solche Dokumentation kann mit einem einfachen Werkzeug aus dem Gebiet des digitalen Gesundheitsmanagements erreicht werden, welches Bildkopien von klinischen Dokumentationsbögen und biologischem Material, das aus dem menschlichen Körper während eines Eingriffs extrahiert wurde, anfertigen kann. Die Sammlung dieser Daten ist sinnvoll, da diese in rechtsmedizinischen Verfahren Wichtigkeit erlangen und auch die radiologische Wissenschaft unterstützen können. Deshalb, wurde Machbarkeit eines solchen Werkzeugs untersucht, mit dem Ziel ein sogenannten positiven 'Beweis des Konzepts' zu führen. Ein solcher wurde in Anlehnung die Standards der 'Integrating the Health Enterprise' (IHE) Vereinigung angestrebt.

Diesen Kriterien entsprechend wurde eine leicht zu verwendende, kostengünstige, nicht aufwendige Anwendung nach dem vierstufigen Prozess der IHE-Vereinigung kreiert und getestet.

Kostengünstigkeit konnte durch den ausschließlichen Einsatz frei verfügbarer, gut vorgetesteter Entwicklungswerkzeuge bei der Erstellung von RISPad erreicht werden. Zusätzlich wurde dem entwickelten System eine preisgünstige hochauflösende Dokumentkammera hinzugefügt, die auf den meisten PCs als simple Webcam erkannt wird. Um eine erheblich wiederholte Softwareinstallation auf verschiedensten Computern in der Röntgenabteilung wo RISPad getestet wurde zu vermeiden, wurde das Programm als server-client-Anwendung bei der der client in einem Standardbrowserfenster läuft konzipiert. Der RISPad-client wurde zudem mit einem klar definierten, einfachen Graphic User Interface (GUI) ausgestattet, das auf die für den Dokumentationsprozess unbedingt notwendigen Funktionen beschränkt wurde.

Das Testen des Prototypen an einer typischen radiologischen Abteilung ergab einen positiven Beweis des untersuchten Konzepts für das RISPad-Projekt. Die weitere Entwicklung von RISPad zu einer professionellen Anwendung ist bereits im Laufen.

Table of Content

1 Introduction

Radiology uses digitized data in nearly all of its diagnostic and therapeutic processes, where the digitalization of the images started already in the late 1970s with the introduction of radiological modalities like CT (computer assisted tomography or computer tomography) and DSA (digital subtraction angiography) [1-3]. Parallel to this, in the 1970s, engineers also started to seek for solutions to permanently store and exchange the newly obtained digital image data for documentation and further diagnostic purposes in human medicine. In this context the idea of a dedicated storage for radiological image data was born, where this storage was implemented as Picture Archive and Communication System (PACS) [4, 5]. Not until the definition of a common standard for storing and exchanging data in the early 1990s the PACS became reality in radiology with the definition of the DICOM 3.0 (Digital Imaging and Communications in Medicine) - standard [6]. Today, a PACS is used in nearly all radiological departments and the modalities and applications storing data in this PACS still follow the DICOM-standard.

With the PACS providing a database for the radiological image data an administrative system hierarchically placed on top of the PACS got necessary. This so-called radiological information system (RIS) manages a work-list of requested examinations and provides this to the various imaging modalities. The RIS may host also radiological service catalogs for the administration of the performed services etc. and could integrate services for generating radiological reports and transmitting them to superior information systems. Accordingly, in hospitals the RIS seldom is stand-alone, in fact it is frequently embedded in larger information environments, the so-called hospital information systems (HIS), which combine all locally available electronic patient data into a consistent electronic health record (EHR).

Though, this framework appears rather permeable for information to be passed unimpededly between PACS, RIS and HIS, it is not. The rather hierarchical setup of this framework, with the PACS placed at the bottom, the HIS at the top and the RIS usually intercalated between the two other systems, facilitates the data-exchange in bottom-up direction, but widely lacks services for handling clinical data top-down.

In consequence, the radiologist normally reporting within the RIS/PACS-environment has no direct access to the health records stored inside the HIS, like scanned copies of external radiological reports, medico-legal relevant information, especially, scanned informed consent sheets and many other. Note that the fragmentary documentation of these documents may quickly gain importance given a still increasing number of medico-legal lawsuits in Austria [7]. Furthermore, clinically and scientifically relevant data, like images of histologic material extracted from the patient, e.g.: a thrombus retrieved from a cerebral vessel [8], cannot be documented directly on site within the RIS/PACS-environment. On the other hand, storing these data in the HIS, again, would prevent the data to be used during radiological reporting in the RIS/PACS-environment.

Obviously, an easy to handle electronic documentation tool able to collect and consistently store all types of information in the RIS/PACS-environment with an option to pass the information to superior health information systems is of great interest. This is particularly true, when not only administrative necessities, but also scientific challenges have to be met. Considering the big variety of possible requirements, which should be manageable by the same software tool, a multi-purpose solution for the task would resemble much of a printer that is virtually able to integrate photographic copies from all kind of documents or materials acquired during a radiological examination or intervention into the existing RIS/PACS environment.

With respect to the described bottom-up architecture of the current health information systems, the most appropriate level to enter the required photographic document-data into the radiological data framework seems to be the PACS. Therefore, the Radiology Information Storage Notepad (RISPad) tool projected and implemented in this thesis was designed compatibly for communication with a PACS. RISPad had to be able to link photographic evidence documents to a corresponding PACS-study. An acceptable quality of the photographic copies of the clinical and medico-legal relevant patient data was achieved connecting RISPad to a high resolution (HR) - document camera device (DCD). Generally, complexity of the components of RISPad had to be kept to a minimum in order to maintain cost effectiveness and to allow the easy integration into daily routine work, thereby using as less hardware resources as possible.

2 Theoretical Background

Experts in the relatively new field of digital healthcare (DHC) have to be inclined to build a bridge between the understanding of information technology (IT) engineers and the needs of health care (HC) – professionals. This claims profound knowledge about HC- as well as IT-standards in order to support the formulation of clinical and technical use cases (CUC & TUC), the rational prototyping of health IT (HIT)-solutions, the appropriate implementation of HIT-tools to reach a proof of concept (PoC) and the assessment of the finally implemented HIT-application in 'real world'-usage.

2.1 Aims of the RISPad-Project

Current radiological HIT-environments usually do not offer a functionality to immediately document clinically and/or medico-legally relevant aspects or documents observed during a radiological examination or intervention. Since these observations and evidence documents are part of the respective examination or intervention, the data should be kept directly within the context of the documentation of the whole examination or intervention.

The RISPad – software tool described in this thesis aims to present a PoC of a DHC-based solution for integrating clinical and medico-legal data directly into an associated modality-specific radiological examination that is stored in an IT-framework typically used at a radiological department. Usually this framework consists of a PACS and a RIS, which most often interact with a superior HIS [5, 9]. All three systems are part of the steadily growing electronic-health (E-Health) enterprise that aims to handle and integrate all sorts of medical data collected electronically in order to constitute a master electronic health record (EHR). This master EHR has to be represented according to accepted HIT-standards to provide a maximum interoperability between different, also standard-conform, HIT-systems. Therefore, on the one hand, a HIT-standard conform implementation of the RISPad-application was sought for the necessary communication with the established HIT-environment. On the other hand, the intended documentation of clinical and medico-legal data had to be easily integrable in the radiological routine work-flow. Hence a web-technology based application capable to transfer photographic images of interventional specimen and clinical evidence documents acquired with a web-HR-DCD to the PACS, was designed and implemented.

In the following a short description of standards and published work-flow profiles potentially relevant for the RISPad-project is given.

2.2 IHE - Integrating the Health Enterprise

Standards of communication and data exchange services used for the so-called integrating the health enterprise (IHE) purpose have been developed during the last four decades. IHE is an initiative engaging IT-developers and –users in order to define, test and implement standards for HIT - processes and communications. The responsible committee in Europe, IHE - Europe, is constituted by national IHE-committees from many European countries, including, e.g., Austria, France or Germany. IHE-Europe supports developers, users and governments concerned with the integration of EHRs and implementations of HIT – applications. Via IHE-Europe accepted standards describing HIT – processes and rules for communication between HIT - systems are published and continuously updated. IHE-Europe also organizes the 'Connectathon', a meeting for users, developers and industrial IT - vendors concerned with development of E-Health software, where interoperability of health information systems can be tested excessively [10].

IHE hosts special Radiology-domain sub-committees, where relevant standard prototype descriptions and standards of communication in Radiology are defined, which both were of interest for the proposed RISPad-project. For the implementation of a standard-conform HIT-application IHE provides a generic four step process that should warrant a maximum of interoperability of the final HIT - solution to existing E-Health information systems. In order to maintain a maximum of interoperability the RISPad-project was aligned to the IHE four step standard process for the implementation of HIT-applications. A short draft of the IHE standard process modified for the RISPad-project is displayed in **Figure 1**.

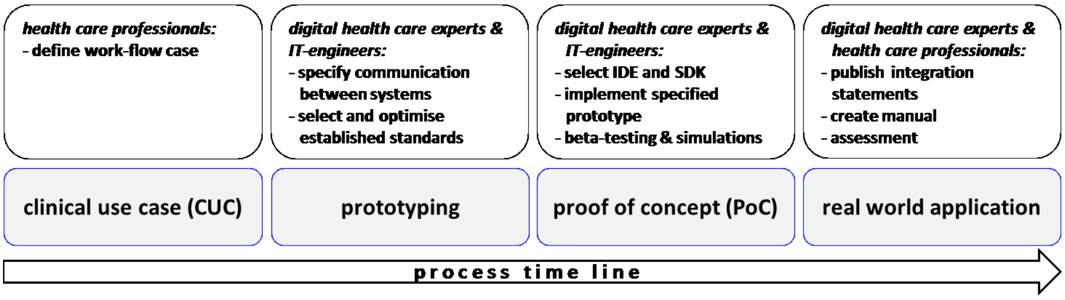

Figure 1: *Modified IHE based process concept for the proposed RISPad-project according to reference: [11]*

Additionally, IHE publishes work-flow profiles for recognised IHE domains, where based on so-called clinical use cases (CUCs) and recommended implementation standards, defined work-flow scenarios are provided. Concerning the RISPad-project, to the best of knowledge, neither a 'final text-stable' - approved nor a 'trial implementation' – tagged profile could be found in the respective IHE-Radiology domain. Thus the kindred, IHE-listed profile: '[WIC] – Web-based Image Capture' was taken and extended with newly defined CUCs and associated technical use cases (TUCs) to create an own RISPad – work-flow profile. CUCs illustrate the clinical work-flow of HC-professionals during the process of image acquisition and administration. Parallel to each CUC an associated TUC describes the same process from a technical point of view thereby assigning the proper components and format-standards to every step of the corresponding CUC.

2.3 IHE-profile: WIC – Web-based Image Capture

The WIC-profile is an IHE-Radiology Technical Framework Supplement that describes the upload of digitally acquired photographic captures and evidence documents to a clinical image management system. Essentially, two CUCs are described in this profile. In *scenario #1* the *Image Modality* is an *Image Capturer* (e.g. a simple digital webcam) that sends digital photographic visual light (VL)-based material to an *Image Management System*, where a new study is created. In *scenario #2* the *Image Capturer* is a graphical evaluation software used to produce digital images generated as secondary captures or photographic VL-images from an existing study, which are sent back to the *Image Management System* [12]. The *Modality* or *Image Capturer*, respectively, and the *Image Management System*, represent the so-called *Actors* in the WIC-profile, which use web-based transmission technology for their communication.

In **Figure 2** the *Actor Diagram* of the WIC-profile showing the web-based transmission of the captured images from the *Image Capturer* to the *Image Management System* is depicted. Note that in this profile the newly added images do not necessarily share the same image data representation, id est, the image data format, with the original raw data images from which they are derived, though, all images are finally stored in the same examination context. The WIC-profile actually describes storage of, for instance, JPEG2000-format files together with binary monochrome format files, where the new images are, also, not derived from the same modality (e.g.: images of a MRI examination and correlated secondary screen captures generated on a tablet-personal computer (PC) are stored together in the same examination context).

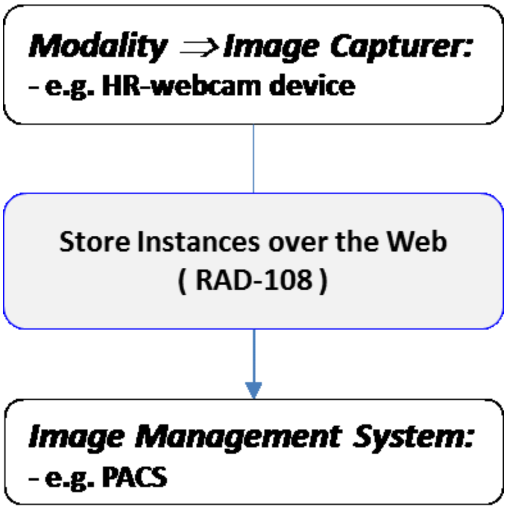

Figure 2: *WIC Actor Diagram modified according to [12]*

Concerning the implementation of the WIC- or a similar profile, one has to keep in mind that all communications require all *Actors* to support at least one option of the *RAD-108* transfer references listed in the WIC-profile. Unfortunately, older PACS applications often do not implement these newer transfer syntaxes, which may cause severe problems in the communication with web-based software implementations. Instead of the standard communication between *Image Capturer* and *Image Management System* that is based on the DICOM binary format, the WIC-profile suggests transmission of files represented in web-technology based DICOM java script object notation (JSON). This is owed to the fact that the WIC-profile assumes that the *Image Capturer* is a web-application implementing its own type of communication with the *Image Management System* thereby using an asynchronous JavaScript and XML (AJAX) -

negotiation. In consequence, this requires the *Image Management System* also to accept certain DICOM transfer options, as stated above, to maintain compatibility with the *Image Capturer*.

2.4 DICOM – Digital Imaging and Communications in Medicine

The DICOM standard provides an object-oriented file format that allows an interoperable data exchange of medical images and documents among various image modalities and image management systems, thereby using a particular DICOM network communications protocol. The DICOM standard was developed by the DICOM Standards Committee that also shares a number of members with the National Electrical Manufacturers Association (NEMA), where latter association holds the copyright of this standard [5, 13].

2.4.1 DICOM - Modalities and - Image Management Systems

In the DICOM-terminology a *modality* means all radiological and, nowadays, visual light (VL) based apparatuses producing digital images, which can be encoded, transferred and stored in the DICOM standard file format. On the other hand, the term *Image Management System* includes all information systems, which are involved in sending, receiving, storing or viewing DICOM standard files. One of these systems is the PACS.

2.4.2 DICOM - Real World Information Model

For the representation of a radiological examination the DICOM standard allots the so-called *Real World Information Model*. According to this model the data of an examination is organised in several levels, which are: *patient*, *study*, *series* and the *DICOM instance* or, simply speaking, the *image object* (**Figure 3**). Every *DICOM instance* contains the specifications of all levels, which are labelled by *unique identifiers* (UID). So every single *DICOM instance* or image is unambiguously attributable to its superior levels and, therefore, finally to the patient it belongs to [14].

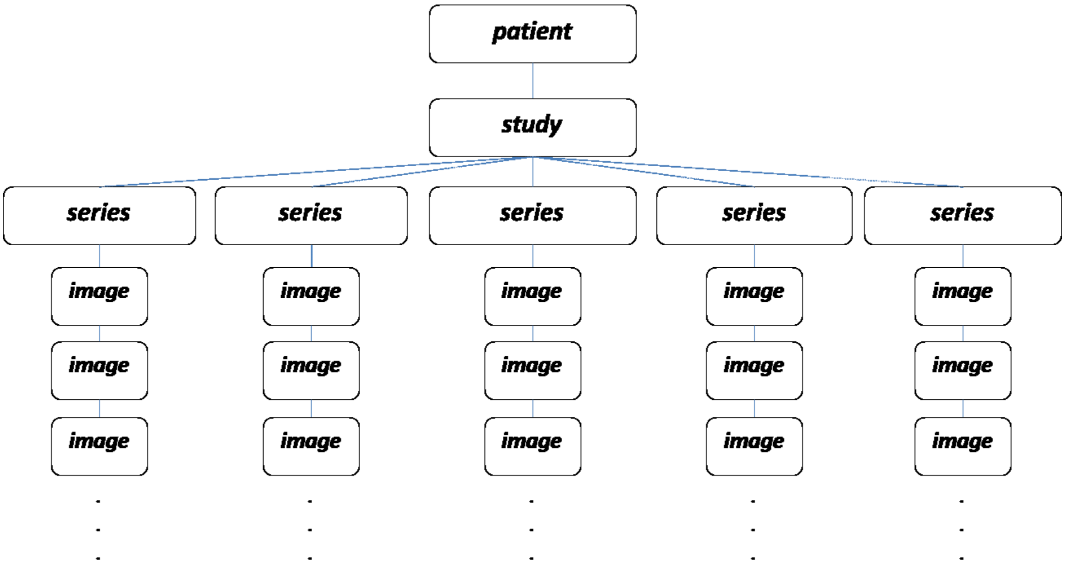

Figure 3: *The DICOM Real World Information Model organises the data of an examination in several levels, which are labelled with UIDs. A full collection of all UIDs are stored within each DICOM object, thereby allowing an unambiguous assignment of each object to all superior levels. Graph modified according to [14]*

Every level of the DICOM *Real World Information Model* is identified by its own UIDs, *Patient IDs* and *Image Numbers*, which are specific for the particular examination and warrants an unambiguousness of all instances, e.g., stored in an *Image Management System*. All UIDs etc. reflecting the hierarchical order of the associated *Real World Information Model* are stored in the DICOM file header in so-called attributes. According to this, there are attributes for the *Patient ID*, the *Study Instance UID*, the Series Instance UID and the Image Number and so on, which enable assignment of each instance to its specific examination. In order to, additionally, specify the appropriate service suitable for a DICOM object, file, the DICOM header also includes a so-called *SOP Instance UID*. The SOP Instance UID denotes the *Service-Object Pair* (SOP) Class that links a *DICOM Service Elements (DIMSE)* to a certain *Information Object Definition* (IOD). For example the DIMSE: *storage* is linked to the IOD: *MR-image*, which instructs an *Image Management System* that a file of this class can be stored in the database [15]. Note that by this DICOM enables an ultimate fast negotiation between a service class provider (SCP) and a service class user (SCU) to find out about their compatibility of semantics for a successful data transfer.

2.4.3 DICOM – file header

In the DICOM file header not only the UIDs of the DICOM *Real World Information Model* are stored in attributes. All the information needed to identify an examination, to interpret or to view an image is also stored as attributes in the header. Attributes according to the DICOM standard follow a certain format. An attribute consists of the so-called *tag*, the *value representation*, the *value length* and the *value field*, where the actual values are stored. The *tag* is a unique identifier composed of a set of two numbers, the group- and the element-number normally written in the hexadecimal system. The *tag* describes the kind of information stored in the respective attribute. The *value representation* determines the value type and its representation in the computer storage. The *value length* informs about the length of the subsequent *value field* that contains the information coded in the respective *tag* (**Figure 4**). The groups are used to embrace attributes of the same context. For example all patient information is gathered in one special group (0x0010), and so on. The element, on the other hand, gives the description what is actually stored in the respective *tag*. For instance, *tag*: (0x0010:0x0010) contains: 'Patient's Name' etc. [16]. Groups are separated into odd and even numbered one. Even numbered groups are defined as public groups, which must contain *tags* defined by the DICOM standard only. Odd-numbered groups, in fact, have to follow a defined standard concerning their declaration, but may contain any content assigned to by the respective developer.

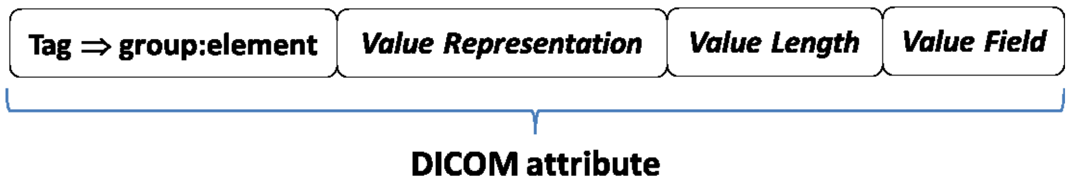

Figure 4: *The DICOM-attribute format*

For generation of a valid DICOM file image instance, besides the image data themselves, at least the declaration of the particular UID-attributes, patient- and vendor-attributes, as well as, image representation-attributes are mandatory (**Figure 5**) [6].

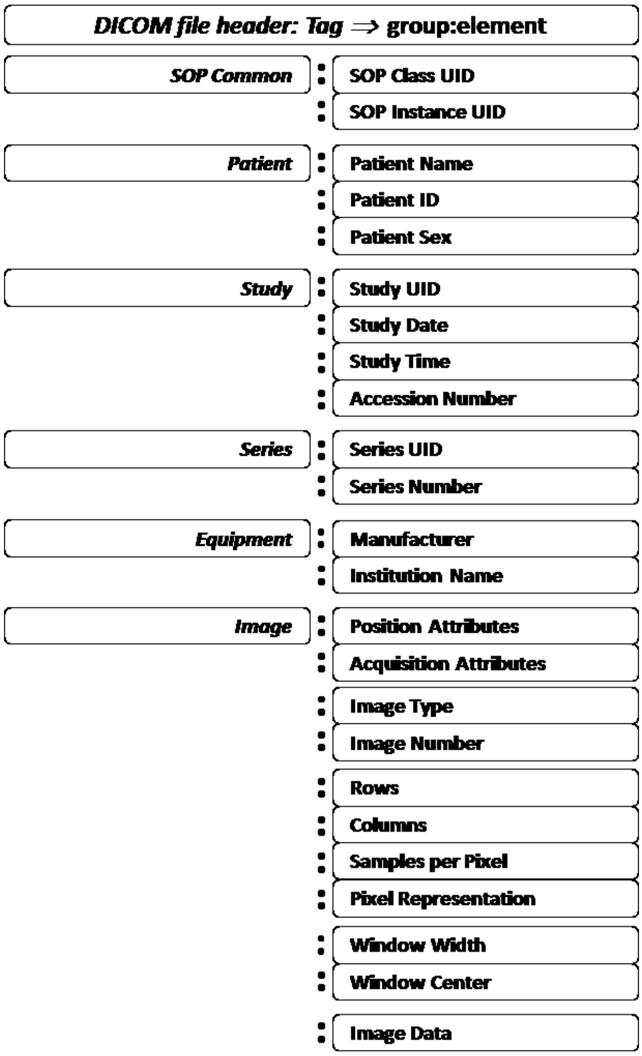

Figure 5: *Mandatory fields in a DICOM file header according to [6].*

2.4.4 DICOM – Native Model and DICOM – JSON Model

While in the DICOM header the information is stored in the DICOM native format, this content may also be represented in other formats, especially when the data is transferred between a service class provider (SCP) and a service class user (SCU), which use web-based technology for their communication. The role of the software package providing the required service classes is usually incurred by the PACS, while the software requesting the respective service class is the connecting modality or, as proposed in this thesis, a special documentation tool. The content of a DICOM-file in the native DICOM-format usually used matches of a multi-part collection of singular XML documents **(Listing 1:** native DICOM model – XML, according to [17, 18]**)**. For the DICOM JSON Model used for web-based communication, this would give a single top-level array of JSON objects **(Listing 2:** DICOM JSON model, according to *[17, 18]***)** [17].

Listing 1: *native DICOM model – XML, according to [17, 18]*

```
<?xml version="1.0" encoding="UTF-8" xml:space="preserve" ?>
<NativeDicomModel>
  <DicomAttribute tag="00100010" vr="PN" keyword="Patient's Name">
    <Value PersonName="1">The Patient</Value>
  </DicomAttribute>
</NativeDicomModel>
…
<?xml version="1.0" encoding="UTF-8" xml:space="preserve" ?>
<NativeDicomModel>
  <DicomAttribute tag="00100020" vr="LO" keyword="Patient ID">
    <Value number="1">123456</Value>
  </DicomAttribute>
```

Listing 2: *DICOM JSON model, according to* [17, 18]

```
  {
     "00100010": {
      "vr": "PN",
      "Value": [ "The Patient" ]
    }
  }
  {
    "00100020" : {
      "vr": "LO",
      "Value": [ "123456" ]
    }
  }
```

2.4.5 DICOM – Pixel Interpretation: MONOCHROME2, JPEG and RGB

The standard photometric interpretation of images generated by most of the radiological modalities is *MONOCHROME2*, which indicates that the pixel values are arranged from low (dark) to high (bright) values in ascending order. Using this scheme the grey scale value of each voxel is coded in a two byte structure, which allows storing all voxel data in a single image plane. While this works fine for most of the radiological modalities producing grey scale values, this does not suffice VL-based image capture devices, which require coding of different color values. Therefore, the *MONOCHROME2* image representation was not suitable for the RISPad-project. RISPad had to be designed to copy various evidence documents or images from interventionally extracted specimen using a color web-HR-DCD [19].

Since the format generated by a web-cam usually is the format of the 'Joint Photographic Experts Group' (JPEG), the DICOM photometric interpretation had to be *YCbCr*, or *YBR_422* in the DICOM standard, when lossless compression for storage of the voxel data is used. Using the *YCbCr* - color model requires three channels or image planes to be managed. The Y-channel regulates the luminance of the image, while Cb- and Cr-channels define its chrominance. Or more rigidly, the Cb- and Cr-channels give the shift of the Y-grey scale value towards either blue-yellow (Cb) or red-green (Cr) chrominance [20]. This requires the *Image Viewer* as well as the *Image Management System* to be capable of managing this format. So far, for the great majority of radiological images the *MONOCHROME2* image representation was appropriate, which is the reason why older PACS implementations often do not support the the *YCbCr* - color model. Though, the DICOM standard provides compatible *Transfer Syntax UIDs (Tag: 0002,0010)* for sending and receiving YCbCr-coded images, an incompatible PACS will refuse acceptance of the data in the initial negotiation, which demands a more basic solution for integrating VL-based image applications [19].

An alternative to the *YCbCr*- interpretation, which does of course not support the compression features of the JPEG-format is the RGB – color model. Again three channels have to be managed: red (R), green (G) and blue (B), where this color model describes every color tone as a mixture of these basic colors. The bandwidth of the color-spectrum in this model is simply determined by the digits available for coding the partition of each base color (e.g.: 256 color-tones reserved for each color channel would require three bytes of storage, where this would allow to code about 17x10^6 color tones a.s.o.) [20]. Thus the base color-tones are usually stored in a three byte structure. Despite the fact that the RGB-model is not the most progressive one, nearly all PACS-implementations accept this format, which should be considered for software developments with integration of VL-based images. In the DICOM-standard the instructions needed in the header to store an instance using the RGB-model is simple. The *PhotometricInterpretation – tag (0028,0004)* in the DICOM-header should read: '*RGB*' with the *SamplesPerPixel – tag (0028,0002)* set to value = '*3'* (id est: one byte per base-color, see above). A relative limitation of this rather simple format is the quite large size of the files of, e.g., HD-resolution VL-based images, since image compression is not provided. As modern PACS implementations store nowadays all image data internally in JPEG2000 lossless compressed format in their database, this apparent drawback of large RGB-files is neglectable. On the other hand, due to the high compatibility of the RGB-format with newer and older PACS-implementations and *Image Viewers* this format is still used often by many applications.

2.5 PACS - Picture Archive and Communication System

With the advent of digital imaging in Radiology also a digital storage for the images became necessary. Simply storing digital image data on a disk or tape with manual administration of the patients rapidly became insufficient as the number of examinations increased significantly. Systems offering a safe durable storing and management of image data in a database were needed. Additionally, as all data were already available in digital format the direct communication between such an *Image Management System* and the various digital radiological *modalities* was required. Not before the development of economically affordable computer systems and the suitable network technology in the 1990s the IT-supported administrating, storing and retrieving of high data-volumes using a computer became possible. The systems had to be able to receive digital image data from the modalities, to associate these images with a certain patient information and to archive and retrieve the data if needed later on, e.g., for viewing images on an *Image Viewer* (**Figure 6**).

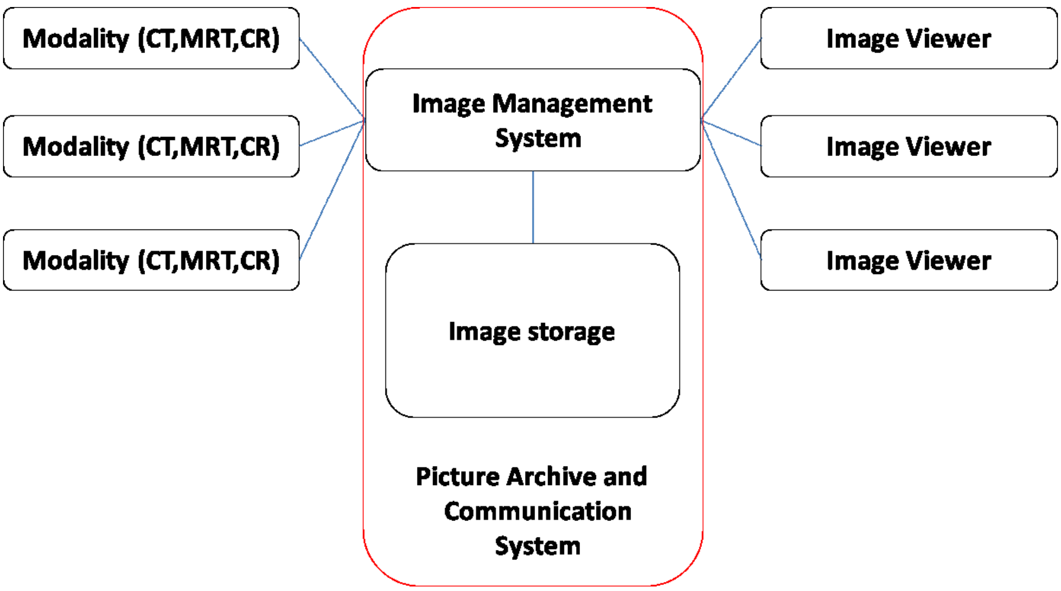

Figure 6: *Schematic sketch of a PACS showing some interactions between modalities, PACS and Image Viewers*

In this way the first PACS applications were implemented, where usually a PACS consists of the *Image Management System* and a mass data *Image Storage*. However, note that the most indispensible prerequisite for these systems was the in parallel developed format standard: DICOM 3.0, which warranted the interoperability of the data generated by the various modalities to be stored in the PACS.

2.6 RIS – Radiology Information System

While the PACS solved many problems concerning the image management, it did not overcome the problem that every patient had to be registered manually on each modality, where an examination was planned. Since this was susceptible to error a supportive IT-system was sought for registering a radiological examination request in a central work-list that also included the patient data provided to each modality of a radiological department. By this the risk to take over erroneous patient data was minimized as the total number of registration steps was reduced to one. At the modality the patient data was linked to the respective images generated during the examination and the whole bulk of information was sent to the PACS. Sometimes, the supportive system could also allow dictating and storing the final report of the examination in its database. Additionally dose data, service lists etc. could also be managed and stored in this system. Since the system does not manage image data, but supports administration of patient- and examination-data, as well as, storing reports and dose information etc., its spectrum of functionality is clearly different from that of the PACS. Therefore, this system was denominated as *Radiology Information System* or *RIS* (**Figure 7**), respectively. In its core the RIS does not only provide the work-list and stores reports and patient data, essentially, the RIS generates most of the patient data and the UIDs later stored in the DICOM header. Note that this requires an already rather complex negotiation between modalities, PACS and RIS.

2.7 HIS – Hospital Information System

An examination request sent to the RIS is usually generated in a superior HIT-system. Naturally, this HIT-system should possess access to all relevant patient data necessary to formulate an examination request. This information is available in the so-called Hospital Information System or HIS, respectively. The HIS hosts all administrative and medical data electronically collected from the various departments in the hospital and manages the interaction between them. In case of Radiology an examination request is generated by the HIS and sent to the RIS. In return, the RIS sends back the study reference and the final report to the HIS (**Figure 7**). However, the exchange of document and image information at the level of the HIS is no longer dedicated to the DICOM-standard alone, which has to be considered when entering data into the various IT-systems.

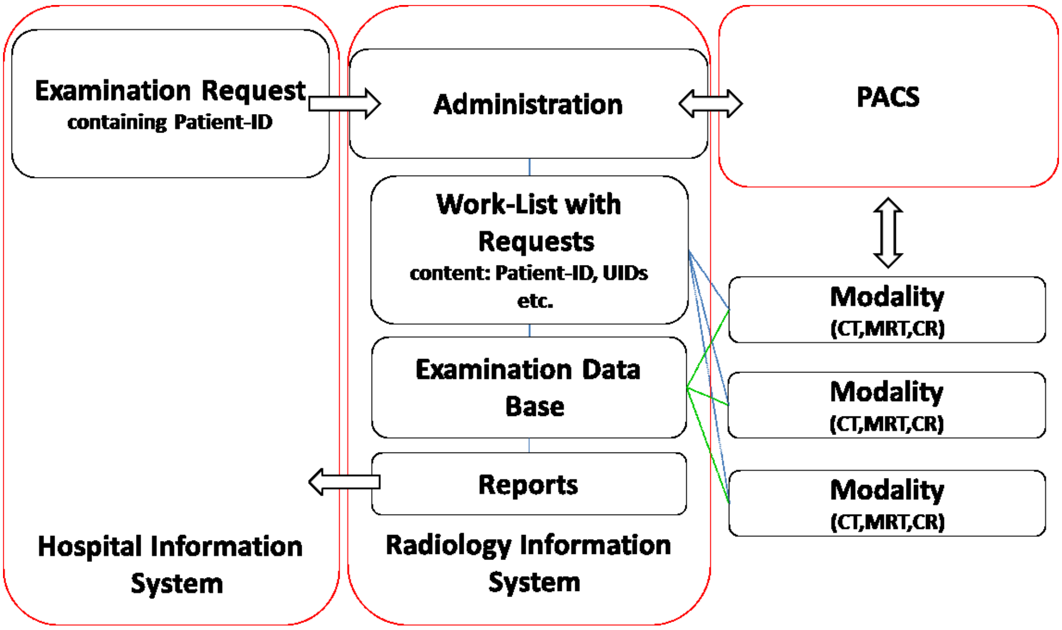

Figure 7: *Schematic sketch of a RIS-implementation showing some interactions with other IT-systems.*

2.7.1 HL-7 CDA Health Level Seven Clinical Document Architecture

Other than the PACS and the RIS, the document standard of a modern HIS is not the DICOM-standard, but the HL-7 CDA standard. This standard is XML-based and structures a document into three levels. The actually clinical relevant information is placed in so-called narrative blocks in level-1, which is purposely not intended to be primarily processed by a computer. Here, for example, the text of a radiological report is placed. Level 2 adds structures to the level-1 blocks in the document (e.g.: a report summary) and level-3 gives further instructions for the appearance of the level-2 structures (e.g.: summary divided into diagnostic report section and section with further therapeutical suggestions). Levels-2 and 3 are both designed to be machine-readable, while, as stated above, this is not true for level-1 content [21]. Sharing these documents between different enterprises requires reverting to the IHE-published standards for 'Cross-Enterprise Document

Sharing', where the document - transfer is based on the 'Cross-Enterprise Document Sharing' (XDS) Integration Profile'. Since XDS does not really depend on the carried content using this standard DICOM-images are also transferable as well as clinical documents [22]. Note that the HIS-RIS dialog is just one small part of the nowadays complex HIS administration (**Figure 8**).

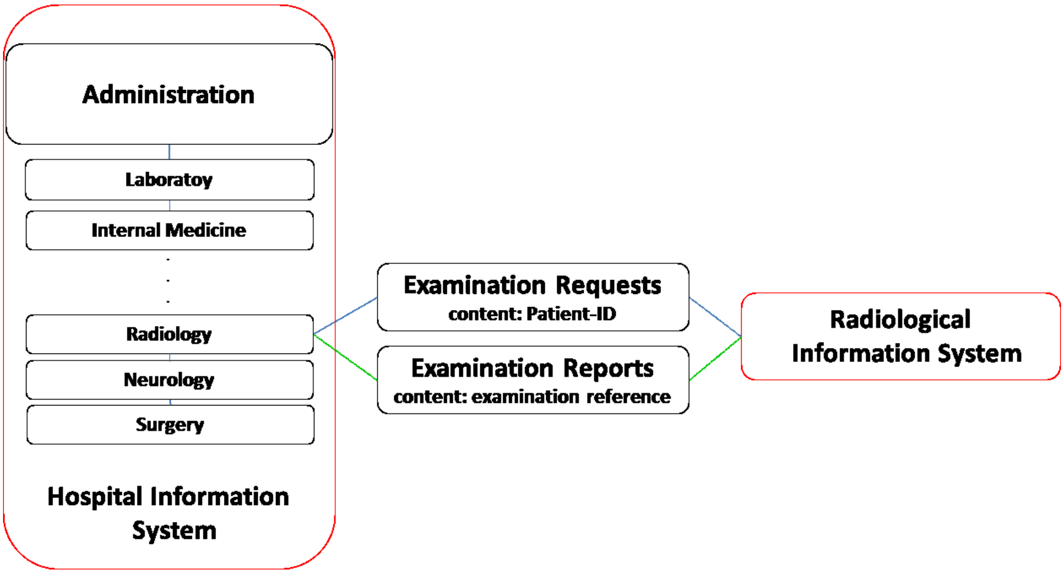

Figure 8: *Schematic sketch of the HIS interacting with the RIS*

2.8 Radiological Work Flow and HIT-system Interactions

The radiological routine work-flow is based on the interactions between the HIS- and the RIS/PACS-environment. Usually, the HIS sends a request for an examination to the RIS, where a work-list for each modality is generated. Simultaneously, the RIS creates/takes over unique patient identifiers later used in the DICOM header and provides this information to the respective modality. After acquisition of the images at the modality, the modalities send certain notifications to the RIS, which are used to change the state of the work-list, to store dosage information etc. Additionally, the modalities send the acquired images to the PACS after the patient identifiers were integrated into the DICOM-header information. In the following the radiologist is able to open an examination at the Image Viewer for reporting controlled by the RIS work-list. The information from the RIS work-list is used by the Image Viewer to query and view the appropriate images of an examination. Reporting of radiological findings can be realised either directly in the HIS, or, as most often implemented, in the RIS environment. In the latter case the final report has to be transmitted to the HIS after the reporting is finished (**Figure 9**).

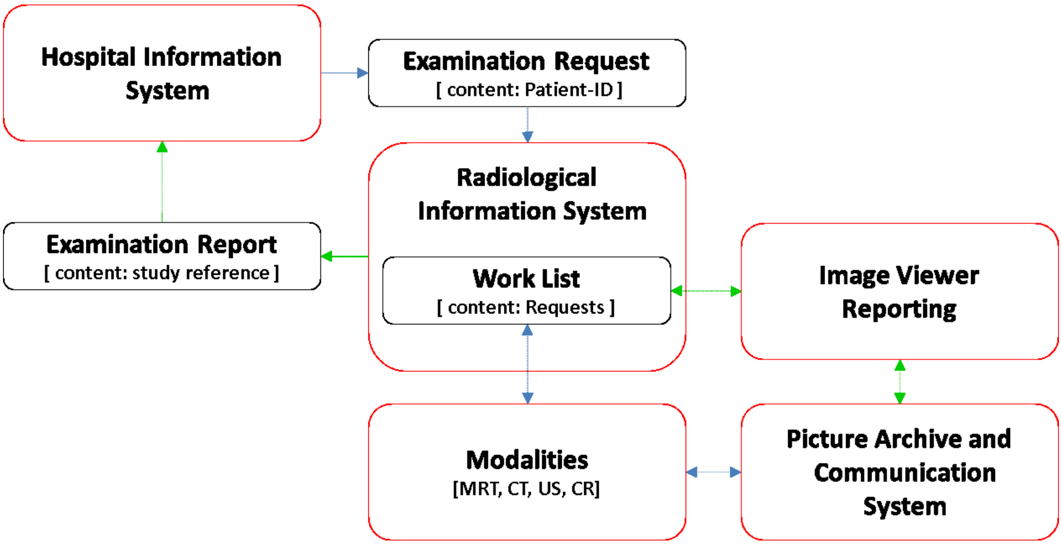

Figure 9: *Schematic sketch of the Interactions between HIS, RIS and PACS*

This system works fine to document a radiological examination and warrants the transmission of the final report to the HIS. However, according to **Figure 9** it is clear that the reporting radiologist cannot directly access any clinical data available in the HIS when reporting is done at the level of the RIS/PACS-environment, because the whole system is strongly orientated in bottom-up direction concerning the general data stream. Since several reports found evidence that the radiologist's access to the clinical records potentially influences the interpretation of image data, this could implicate a potentially serious source for documentation errors. Note that the great majority of RIS/PACS-workplaces do not possess full (if any !) integrated HIS-functionality, which often deprives the radiologist from relevant clinical data [23].

Vice versa, relevant information obtained at the level of the RIS/PACS-environment cannot be directly integrated into the patient’s record file at the level of the HIS by the radiologist. In this setting the possibility of obtaining data, other than image data, at the level of the PCAS or RIS, is not available. For instance, the radiologist has no possibility to document any informed consent talk or script. This is still done manually using paper-ware with the 'promise' to copy an informed consent into the HIS, which cannot be controlled by the person who actually is responsible. In the same way it is not possible to document any relevant clinical data together with the images in the PACS. In the first case it is possible that an informed consent sheet is lost on its way to get copied to the HIS. This means higher risks for the patient, as important informations about pregnancies, incompatible implants or adverse drug reactions could be potentially lost. Moreover, in case of legal disputes clarification of a claim is hardly possible, when the informed consent sheet is missing. Furthermore, the inability to directly store clinical findings relevant for an examination together with the images could hamper documentation in accordance with rules of GCP [24]. Thus an instant documentation of any radiologically relevant data at the level of the RIS/PACS-environment together with the image data with direct access to these data by the radiologist is desirable. In this context is was shown that the automatic, immediate reporting of technical procedure details in Radiology was significantly less prone to documentation error than the delayed manual inserting of procedure details [25].

Although the implementation of electronic health records in the HIS is rather complete, nowadays, in most of the hospitals using electronic documentation facilities, the interface performance between the HIS and the RIS/PACS-environment concerning clinical data is only modest, at the best. The historical background for this may be seen in the fact, that traditionally clinicians were used to pose just a specific question for the radiologists, when requesting a radiological examination. This quite reflects the current common negotiation between the HIS and the RIS, where the radiologist does not gain much direct access to the electronic health records stored in the HIS, although this dialog has been already recognised as in principle highly complex [9, 26-28]. One way to overcome at least some aspects of this problem could be placing the radiological reporting functionality at the level of the HIS, which already led to questioning the future use of a separate RIS per se at all [29]. Anyway, the need of a full interoperability of the HIS-RIS-PACS-applications becomes most evident in the context of patient centered electronic health record (EHR) systems like the 'Elektronische Gesundheitsakte' (ELGA) [30-32].

Although, recent developments for the administration of EHR aim to fully integrate all types of patient information into comprehensible systems, e.g. Austria's electronic health record project: ELGA, this project still relies on the traditional bottom-up concept, which is found in most EHR-implementations during the last 40 years. Bottom-up orientated systems do not consider most of the daily needs of medical and para-medic personal engaged in public health care as these tools were designed for public healthcare management and economics [33]. Please note that these systems represent ideas of EHR-design, which were in part generated even before a personal computer became available in daily routine work ! Considering the technical possibilities currently available to support efforts in the wide field of digital health care, the presented project aims to realize a practical tool that enables incorporation of correlated clinical data directly into radiological examinations. Thereby valuable data, which is not obtainable with established facilities today, is generated suitable to be used further as information data base in clinical decision support systems [27].

As for the part of the proposed thesis project, it is therefore most evident that the development of an assistive health informatics tool that allows adding clinical data encapsulated in standard-format RIS/PACS objects to a radiological examination is of utmost importance [6]. Such a tool could also facilitate clinical research and could help to prevent radiological reporting error [25, 34]. Since the intention of the proposed project is to use the RIS/PACS as an entry-point and to strictly to keep up with defined information technology standards, the integrity of involved information systems was not to be affected and an absolute compatibility of the added data with the established bottom-up data-structures of the HIS-RIS/PACS environment had to be warranted. However, the feasibility of such a concept has been already shown for digital photographs stored along with patients' radiological image data [35].

3 Methodology

RISPad was implemented as a Node.js® server network application with the client accessible via a standard web-browser connected to a HR-DCD that is able to capture high resolution images from evidence documents and clinical material. The implementation of a VL-based interface, to efficiently record and store clinical data sets together with their corresponding radiological examination in the PACS was conceptualised. The interface had to function like a notepad where relevant, non-radiological image data records could be collected during an examination and made directly accessible in the PACS. Simply scanned hand-written data or photographical documentations of medico-legal query-sheets had to be storable together with the correlated radiological image data within the same DICOM-study. In order to warrant a frictionless integration of the proposed software application into daily routine the whole process was first described by appropriate use cases. Then the various program components of the RISPad application and their implementation were defined. Finally, the necessary hardware and appropriate software development tools were specified.

According to common definitions of product requirements in digital health care, (1) the practicability of the proposed tool in daily routine, (2) its conformance with the effective legal background, (3) its application in scientific questions under conditions of Good Clinical Practice (GCP) and (4) its stability within the RIS/PACS environment were considered [6, 24].

3.1 RISPad Use Cases

The RISPad CUCs were defined based on near miss incidents derived from risk management documentations and law-suit cases regarding radiology routine work-flow. Near miss incidents denote situations in clinical routine work-flow, which threatened a patient's health or safety, but could just be obviated. Accordingly, these incidents can be considered of high clinical relevance. Two scenarios were identified, where in the first scenario the patient received a diagnostic or interventional examination and in the second scenario an intervention with extraction of a histological specimen was performed on the patient. Though the CUCs and TUCs appear relatively identical, it is important to note that the image quality of the HD-CDC is required to allow a sufficient macroscopic evaluation of the specimen material composition and to provide excellent readability of copies form evidence documents. Since the documentation of evidence documents or bioptic material should be possible everywhere in the radiological department, an easy access to the RISPad application with minimal hardware requirements had to be warranted.

3.1.1 Specifications

As the environment the DICOM-based communication in the Department of Radiology in the University Hospital Tulln was specified. The documentation was considered according to the IHE-WIC-profile, where using a web-based HR-DCD images from interventional specimen and evidence documents were taken and linked to their correlated examinations, thereby keeping up with the structure of the *DICOM-Real World Information Model (RWIM)*. Webcam acquired images are stored in the first instance in JPEG-format locally. For transmission of image data within the web-application Base64-coding was used to ensure code page-independent readability of the transmitted data. The role of RISPad was considered as an *evidence document web-server* (EDWS) that was realised by a web-server application that used a WebSocket based real time bi-directional event driven communication for the negotiation with its web-clients, latter acting as *evidence document web-clients (EDWC)*.

3.1.2 Use Case #1: Amendment to Informed Consent for a Radiological Intervention

Clinical Use Case: An informed consent sheet explaining a planned magnetic resonance tomography (MRT) examination is handed out to a patient at the ward. After filling in all questions with the medical assistant, where no contraindication is found, the patient is brought to the MR-unit. The radiologist checks all questions again, but no conspicuousness is revealed. The patient is brought into the magnet, where a strong image distortion is noted at the survey. As the patient suddenly feels a severely stabbing pain at his left cheek he immediately is retrieved from the magnet. An x-ray shows a metallic splint in the left lower eyelid. Asked for his profession it turns out that the patient is a metalworker who did not think of small injuries, which were at that time without any further consequence, when he previously answered the informed consent sheet. The incidence is noted on the informed consent sheet and the sheet is documented together with survey in the PACS by the radiology technologist (RT). For documentation the informed consent sheet is copied using a high resolution webcam, which is connected to a software application that displays the copied images and allows selection of the correct examination, where the images are linked to in the following.

Technical Use Case: An examination request is generated by the *HIS* and transmitted to the *RIS*. In the *RIS* the respective examination is specified and inserted into the modality work list. The modalities fetch the request from the RIS-work list and acquire the appropriate images, which are sent back to the PACS, after a *DICOM-RWIM* of the current examination is built. A *notification image* (id est: one example DICOM-image of the original examination generated by the respective modality) that contains the patient specific *DICOM/PACS entries* of the specified examination is sent to a SCP to cooperates with the *EDWS*. The EDWS adds the patient and examination specific information read from the notification image (NI) to its *documentation work list*.

The *documentation work list* is provided to several *EDWCs* throughout the department. Using a *HR-DCD* connected as webcam to each *EDWC*, a series of *evidence documents* (EDs) is copied and locally stored in the *EDWC* application. Then the corresponding examination is selected from the *documentation work list* and linked to the ED-series at the GUI of the *EDWC*, thereby creating a collection of evidence documents linked to the associated *documentation work list* entry that warrants correct association of the *ED-series* to the corresponding examination stored in the *PACS* later. This *linked collection* is summarized in a so-called *tie*, which describes the link between the evidence documents and the specific examination data.

The *tie-summary* is sent to the *EDWS*, where it is translated into a data object called *tie-structure* that is convertible into a DICOM-series consisting of RGB-formatted files to guarantee down-compatibility with older PACSs. After converting the tie-structure into a DICOM-series this series is sent to the *PACS*, where it is stored according to the specified, previously linked examination (**Figure 10**).

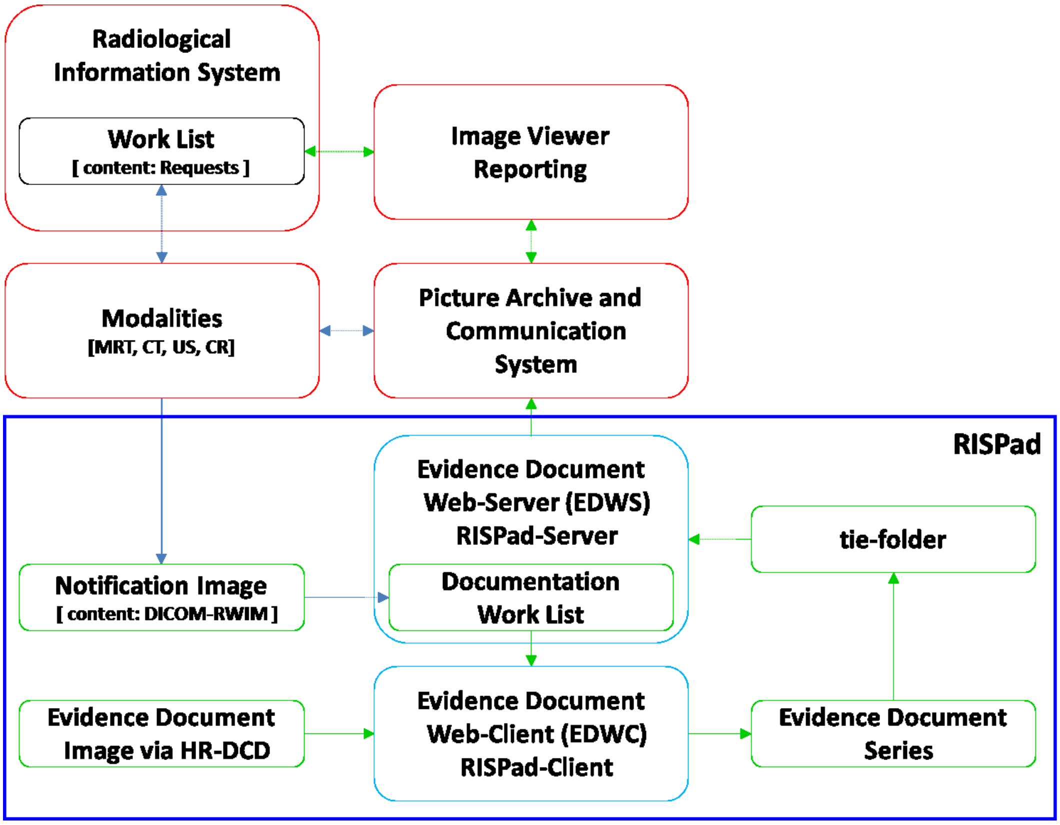

Figure 10: *Schematic sketch extending **Figure 9** showing the interactions of the planed RISPad-project within the RIS-PACS environment. The HIS connectivity was neglected for clarity reasons.*

3.1.3 Use Case #2: Documentation of a Specimen Extracted During A Radiological Neurointervention

Clinical Use Case: A patient is brought to the radiological department with acute ischemic stroke. MRI reveals eligibility of this patient for neurointerventional thrombectomy. During this procedure at the angiography room at the radiological department a 4 cm thrombus composed of different material types is extracted from a major brain vessel and the cerebral circulation is successfully restored. The involved RT documents the images of the angiographic series at the modality and, also, documents the extracted thrombus for forensic and scientific purposes using a high-resolution webcam. The images series generated with the webcam device are stored together with the angiography examination.

Technical Use Case: Within the *HIS* a request for an angiography is compiled and transmitted to the *RIS*, where the examination is inserted into the modality work list. The modalities fetch the request from the RIS-work list and acquire the appropriate images, which are sent back to the PACS, after the specific *DICOM-RWIM* of the current examination is built.

A *notification image* that contains the patient specific *DICOM/PACS entries* of the specified examination is sent to an *EDWS* that adds the contained information to its *documentation work list*. The *documentation work list* is provided to the *EDWC* onsite at the angiographic unit.

Using a *HR-DCD* connected as webcam to the *EDWC*, a series of images depicting the extracted material is produced and locally stored in the *EDWC* application. The appropriate patient specific *DICOM/PACS entries* are selected and linked to the image-series at the *EDWC*, thereby creating a linked collection that already incorporates the suitable patient specific *DICOM/PACS entries* to warrant correct association of the image-*series* to the corresponding examination stored in the *PACS* later.

A structured summary of the linked collection, a so-called *tie* is sent to the *EDWS*. The *EDWS* generates an object called *tie-structure* that can be used to create a series of DICOM-compatible RGB-format files linked to the specified examination, which is finally sent to the *PACS* (**Figure 10**).

3.2 RISPad Software Components

For the prototyping the project was structured into (1) the *RISPad - background services* managing the complete RISPad-PACS negotiation and image translation, (2) the *RISPad-server* handling the communication with the various clients and (3) the *RISPad-work place* including the *RISPad-client* and the HR-DCD acquiring all the evidence documents and images of the specimen (**Figure 11**).

RISPad was planned to be embedded in an existing scientific network environment, where parallel to RISPad also other tasks have to be performed 24 hours a day. Additionally, a light weight, low budget solution was searched that should require minimum efforts to be installed in the running routine work at a radiological department. RISPad work-flow had to be simple and robust thereby offering an ergonomic graphic user interface (GUI) in order to widely warrant the correct linking of a RISPad-document series to its corresponding examination in the PACS. A predefined storage directory, the so-called tie-folder, was used as common platform to temporarily store data for the file exchange between the various RISPad-components.

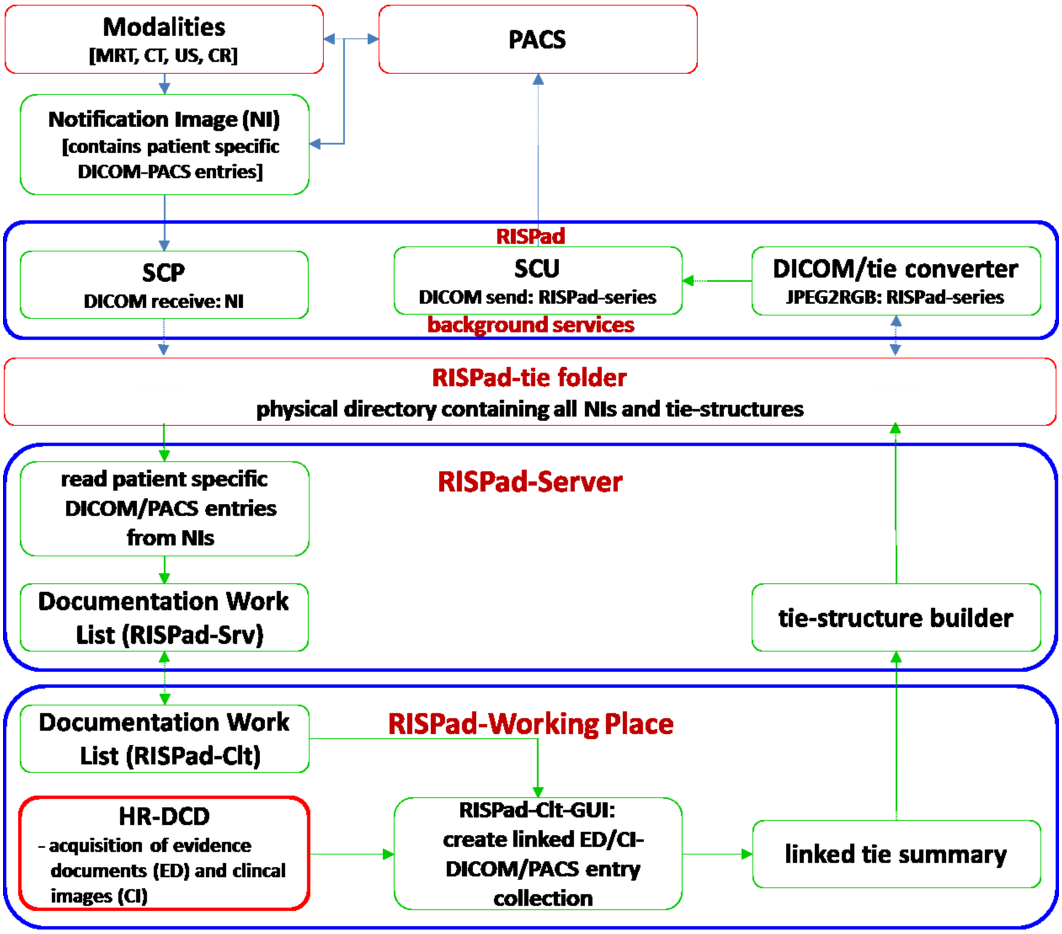

Figure 11: *Schematic sketch of the final modular structure of the RISPad-project. Thick blue line –boxes mark the major software components of the RISPad-project: (1) RISPad-BkS, (2) RISPad-Srv and (3) RISPad-WkP. The RISPad-tie folder is used as a central platform to exchange data between RISPad-components, modalities and the PACS.*

3.2.1 RISPad - Background Services (RISPad-BkS)

RISPad-BkS is a collection of a SCP, a SCU and a RISpad-specific DICOM/tie-converter. All components are permanently working background processes, which continuously control their dedicated functionalities.

The RISPad-BkS-SCP is conforming to a standard SCP that provides all classes of DICOM-transmission services needed to receive the specified *notification images* from the respective modalities existing in a radiological department. The received DICOM-files are stored in a preassigned storage directory, the so-called tie-folder. As RISPad-BkS-SCP the standard C++-based SCP used for several years at the department of Radiology was planned.

The RISPad-BkS-SCU is activated by the RISPad-BkS-DICOM/tie converter after a complete RISPad-DICOM-series of evidence documents and/or clinical images is successfully built. It directly connects to the PACS and sends the data provided by the RISPad-application. The SCU was specified as a JAVA-written application that represents the capstone process of the RISPad documentation.

The RISPad-BkS-DICOM/tie converter was defined as a JAVA-written multi-threading process set as daemon permanently surveying the RISPad-tie folder for newly entered RISPad-tie structures placed there by the RISPad-Srv. It handles the translation of originally JPEG-formatted document images from the HR-DCD into RGB-formatted DICOM-files and is responsible for the conformity of the DICOM headers created by this software. The software should be optimised for immediate work-up of RISPad-tie structures using synchronisation of non-atomic image translation routines.

3.2.2 RISPad – Server (RISPad-Srv)

The RISPad-Srv was designed as an event driven application with the ability to concurrently handle multiple connections. It is responsible for read-out of relevant patient and study specific DICOM-tags from *notification images* placed in the RISPad-tie folder by the RISPad-BkS-SCP [36]. The server constitutes an internal server data-base, the so-called RISPad-Srv-documentation list, that contains all patient specific DICOM/PACS entries read from the NIs. The DICOM/PACS entries placed in the RISPad-Srv-documentation list are provided to the RISPad-Clts.

3.2.3 RISPad – Working Place (RISPad-WkP) and RISPad – Client (RISPad – Clt)

RISPad-WkP was specified to run an html5- and JavaScript-written RISPad-Clt application in a standard web-browser as provided by the RISPad-Srv. The RISPad-Clt adopts the HR-DCD as webcam device and stores the documented images in the local random access memory (RAM) storage. Via the RISPad-Clt-GUI the user should be able to control the acquisition of evidence documents and clinical images, to manage the link to the corresponding examination and to access the documentation work list. RISPad-Clt also implements access control triggered by a password.

3.3 RISPad Hardware Components

3.3.1 High Resolution – Document Camera Device

As HR-DCD a low cost document camera (IPEVO®, Ziggy HD-plus®, Sunnyvale, CA, USA) providing a resolution of 8,0 Megapixel that is connected via a universal serial bus (USB) plug to a standard-PC was used. The camera is recognized as webcam by standard operating systems (MAC-OS®, Windows®, Linux [tested release: Ubuntu 12.04]) implementing the USB 2.0 video class (UVC) interface, where direct access via a standard web browser (minimal requirement: ie 8) is possible. The camera offers a high frame rate (up to 30 fps), which allows live streaming at a resolution of 1920 x 1080 pixels. Fast complex and single focus fields enable rapid focusing of the captured images. The highest resolution possible is 3264 x 2448 pixels (at 15 fps), which allows focused image captures with sufficiently high resolution for all purposes required by RISPad (**Figure 12**).

Figure 12: *Full view capture (3264 x 2448 pixels) of thrombotic clots extracted from the middle cerebral artery of a patient with acute stroke.*

The optical system operates a full autofocus lens combined with a 12x digital zoom. Minimal object distance is about 150 mm, which offers the full resolution at a viewing field of app. 200 x 200 mm. Using a height extension the available viewing field becomes about 492 x 365 mm, which applies to size standards of

DIN A3. Both viewing field adjustments allow best readability of texts and excellent detail differentiability, which was tested excessively prior to the integration of this hardware component into the RISPad- environment (**Figure 13**).

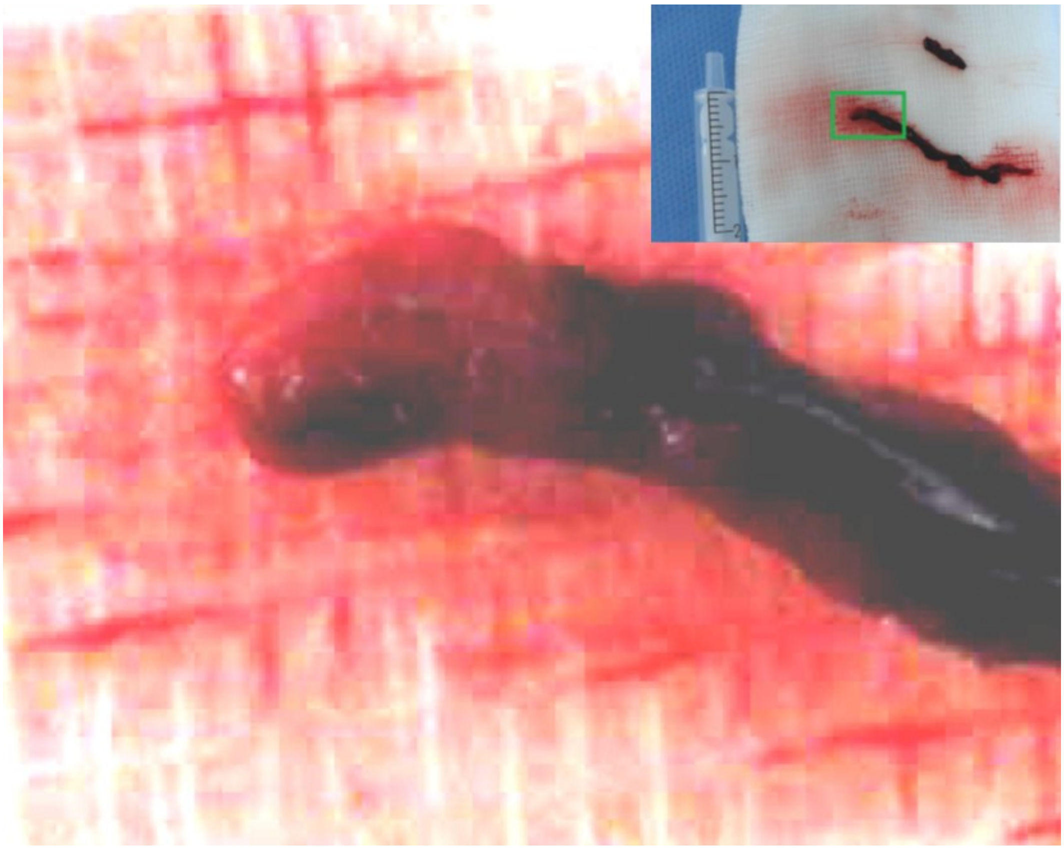

Figure 13: *Magnification of a 20 x 10 mm detail (green marked area in the guide view in the left upper corner) showing the different composition of the thrombus depicted in **Figure 12**.*

3.3.2 Personal Computer and Network

3.3.2.1 Server-PC

The RISPad-server was implemented on an Intel® Core™ i7-4770k PC (4 physical cores) with 8.00 gigabytes on board random access memory (RAM). Microsoft Windows® 7 and 8.1, respectively, were used as operating systems in order to stay compatible with the equipment established at the department of the planned installation.

3.3.2.2 Client-PC

Either low cost Intel® Core™ i3-6100T systems (2 physical cores 3,2 GHz, 3 MB Cache, 35 W, 4 Threads) or so-called thin-clients in a virtual desktop infrastructure as provided at the department were used.

3.3.2.3 Network

Server and client PCs were connected to the in-house 10/100/1000 MBit network infrastructure. All client-PCs were located within the same network segment. The server-PC was able to directly connect to the various imaging modalities and to the PACS.

3.4 Software Development Kits (SDK) and Integrated Development Environments (IDE) Used Implementing RISPad

3.4.1 HTML5 – Programming Language

HTML (hypertext markup language) in its 5th version is currently the most recent core-language of the World Wide Web. HTML5 was extended by several options like video, audio, RAM-management and graphical abilities, all of them useful in the implementation of the RISPad-application. Format control in HTML5-written applications is usually managed by Cascading Style Sheets (CSS). HTML5 and CSS are supported by most of the popular web-browsers. Together, HTML5 and CSS structure and manage the appearance of a web-application. The main script of the RISPad-Clt was written in HTML5.

3.4.2 JavaScript – Programming Language

JavaScript (script programming language) extends the possibilities of HTML5 and CSS by introducing the possibility to dynamically alter and refresh the behavior of a web-page. Code written in JavaScript is interpreted by most of the popular web-browsers, though some differences between the various browsers have to be considered. Using JavaScript, a developer can revert to many useful program libraries, where one of the most popular ones is jQuery. The jQuery plug-in: 'DataTables' was used to implement the document work list in the RISPad-Srv and RISPad-Clt [37]. Additionally, the plug-in: 'webcam' was used for RISPad-Clt to gain access to the HR-DCD, which could then be controlled like a simple webcam-device [38]. Additionally, the package: 'auth-connect' was used to control the user access to the software and the package: 'daikon' allowed easy manipulation of DICOM-files [39]. Major parts of the RISPad-Clt and the complete RISPad-Srv were both coded in JavaScript.

3.4.3 Node.js® – Server Side JavaScript Environment

Node.js® is a server side event driven asynchronous runtime implementing JavaScript as its command language. The developers state a close relation to the V8 JavaScript engine built into the Chrome web-browser published by Google®. The I/O-model provided by Node.js® is described as '… an event-driven, non-blocking I/O model that makes it lightweight and efficient…' [36]. A great advantage of Node.js® is its big program library consisting of numerous routines, the so-called packages, which offer practical solutions for nearly all problems encountered during design of a software-project. Down-load and installation of the packages in the Node.js® environment is managed by the node package manager (NPM). The RISPad-Srv/Clt – network application was implemented within the Node.js® environment.

3.4.4 Java – Programming Language

Java® (programming language) is a registered trademark of Sun Microsystems®, now owned by Oracle®. Originally referred to as 'oak' the name 'java' was placed instead. Java® is a very popular, freely available object-oriented language that intends a certain independency from the used operating system. Therefore, code written in Java® is just precompiled and interpreted later at runtime by an operating system specific virtual machine running as background process on most computers today. Java® allows multi-thread programming, where threads may be declared to run independently from a main function as so-called daemons. Performance control of different cross-armed threads is achievable by synchronisation of specified code blocks that forces the block-wise execution of a thread in order to provide a certain result to entangled thread. The RISPad-BkS-SCU and the RISPad-BkS-DICOM/tie converter were implemented as cross-armed threads declared as daemons with code synchronization using the Java® SE 8 version. For handling of DICOM processes an external library was used (DCM4CHE® 3.3.9 [40]).

3.4.5 NetBeans Software Development Kit

NetBeans® is a freely available - IDE sponsored by Oracle® that provides a high performance SDK for Java® also including comfortable debugging facilities [41]. Additionally, plug-ins for Node.js-, JavaScript and HTML5 are available for software development, which offer optimized run-time testing and debugging [42]. For RISPad development NetBeans® IDE 8.0.2 was used.

4 Implementation / Testing / Evaluation Results

4.1 The RISPad-tie

The central storage and communication structure of the RISPad-application is the RISPad-tie. It represents a directory-like structure that contains: (1) the notification file sent to the tie directory, (2) the JPEG-images acquired by the RISPad-Clt and, after translation, (3) the converted RGB-DICOM images. The converted RGB-DICOM incorporate the PACS-relevant DICOM entries of the patient's examination derived from the notification file and the image information gained by the RISPad-WkP. The RGB-DICOM images of a tie get new entries for the series and image instance UIDs as well as for the image representation, which is RGB. The rest of the DICOM-RWIM entries of the respective examination are kept in order join the new RISPad-series to the existing specific patient study in the PACS. The tie-name contains the patient name and ID, as well as the acquisition date and time (down to milliseconds). Additionally, the IP-address of the sending RISPad-Clt is noted. This was introduced to guarantee unique ties on the one hand, and to increase network and image information security on the other hand. All entries of the tie also become part of the DICOM-header to warrant traceability of the documenting persons. Please note, that the RISPad-Clt can be started only on PCs with personalized accounts.

The proposed structure was chosen because in this way RISPad-ties can be handled comfortably as objects inside the application, while a direct physical representation of these objects as a directory with single files in the tie-directory is also possible. The latter feature can be used for documentation purposes and allows further process control.

4.2 Implementation of RISPad-BkS

The RISPad-BkS-DICOM/tie converter was implemented in Java®-language. Java® allows multiple threads within the same application. A thread set as 'daemon' has the ability to run and stay alive independently from its main-function. During the program execution the evaluation of threads is cyclically placed on hold and calculation time is given to the next thread. Only atomic-functions warrant complete result calculation, while non-atomic process may temporarily stop before a result needed in a cross-linked thread is calculated. This would also block the cross-linked thread. Since nearly all functions are non-atomic complete evaluation of most functions is not warranted in multi-threading, which may cause problems with event handling. Using synchronized code blocks within the program helps to avoid adverse program states due to non-atomic function results. Synchronization of certain code blocks was implemented in the RISPad application. To control the daemon a possibility to stop the thread by resetting the thread-state has to be considered. Therefore, a function that can be called from outside the daemon, e.g. by the main function, has to be defined. The control loop of the respective daemon is then placed in the run-function that is usually overridden by the code needed for in the thread. In the snippet given in **Listing *3*** a recursive directory observer is called in the declared thread that is previously declared as daemon.

Listing 3: *Snippet showing a thread of the RISPad-BkS-converter that defines the demon observing the tie-directory. The code was truncated. (Java®-language code)*

```
    //constructors of ScanDirThread
    ScanDirThread ( String XDir, String Sequ, boolean bDaemon ) {
        fDir    = new File(XDir);
        strSequ = Sequ;

        setDaemon(bDaemon);//define thread as daemon
        state = fDir.isDirectory();

        App.frameMainWindow
           .stdTTPsetSystemText( "daemon > check state: "
                                 + String.valueOf(isDaemon()) + "\n\r"
);
        App.frameMainWindow
           .stdTTPsetSystemText( "daemon > check directory: "
                                 + String.valueOf(state) + "\n\r" );
    }


    //methods of ScanDirThread
    public void stoppen() {
        state = false;
    }

    @Override public void run() {
        //variables & objects
        int i;
        ArrayList<File> filList = new ArrayList<File>();

        //to do
        //further code follows here
    }
```

JPEG images from the RISPad-Clt use Base64 coding for the image transmission. These were directly read into a BufferedImage object in the RISPad-converter, where all RGB values could be easily extracted by the .getRGB method and transferred to an RGB-image buffer (**Listing *4***).

Listing 4: *Snippet showing the extraction of RGB-values from a BufferedImage object using the .getRGB method (Java®-language code)*

```
colMat= new byte[ (bufJpg.getWidth()*bufJpg.getHeight()*3) ];
for( x=0; x<bufJpg.getWidth(); x++ ) {
   for( y=0; y<bufJpg.getHeight(); y++ ) {
       colJpg = new Color( bufJpg.getRGB(x,y), true );
       xy = ( y * bufJpg.getWidth() ) + x ;
       colMat[(xy*3) + 0] = App
                             .frameMainWindow
                             .outByte(colJpg.getRed());
       colMat[(xy*3) + 1] = App
                             .frameMainWindow
                             .outByte(colJpg.getGreen());
       colMat[(xy*3) + 2] = App
                             .frameMainWindow
                             .outByte(colJpg.getBlue() );
   }
}
```

After construction of the new DICOM-images, which contain the patient and examination specific tags the DICOM-RGB representation tags are added to the DICOM-RGB-object (**Listing 5**).

Listing 5: *Snippet showing the 'remodelling' from initial MONOCHROME2-pixel interpretation to RGB-interpretation in the DICOM-prototype header by the RISPad-demon (Java®-language code)*

```
protected void RGBcreate () {
        Header.putInt( Tag.SamplesPerPixel,
            Header.vrOf(Tag.SamplesPerPixel),
            3 );
        Header.putString( Tag.PhotometricInterpretation,
            Header.vrOf(Tag.PhotometricInterpretation),
            "RGB" );
        Header.putInt( Tag.PlanarConfiguration,
            Header.vrOf(Tag.PlanarConfiguration),
            0 );
        Header.putInt( Tag.PixelRepresentation,
            Header.vrOf(Tag.PixelRepresentation),
            0 );
        Header.putInt( Tag.BitsAllocated,
            Header.vrOf(Tag.BitsAllocated),
            8 );
        Header.putInt( Tag.BitsStored,
            Header.vrOf(Tag.BitsStored),
            8 );
        Header.putInt( Tag.HighBit,
            Header.vrOf(Tag.HighBit),
            7 );
        Header.putString( Tag.SOPClassUID,
            Header.vrOf(Tag.SOPClassUID),
            "1.2.840.10008.5.1.4.1.1.4");
        Header.putString( Tag.InstanceCreatorUID,
            Header.vrOf(Tag.InstanceCreatorUID),
            "1.2.40.0.13.1.1");

        Header.remove( Tag.SmallestImagePixelValue );
        Header.remove( Tag.LargestImagePixelValue );
        Header.remove( Tag.WindowCenter );
        Header.remove( Tag.WindowWidth );
        Header.remove( Tag.WindowCenterWidthExplanation );
return;
}
```

The finally configured DICOM-RGB-objects are used to initiate a work-up by the RISPad-BkS-SCU that, after converting the objects to real world DICOM-files, sends the DICOM-images to the PACS (**Listing *6***).

Listing 6: *Snippet showing DICOM-send performed by the RISPad-BkS-SCU using the library dcm4che (Java®-language code)*

```
//start dicom negotiation
dcm_send.configureTransferCapability();
try {
  dcm_send.start();
 catch (Exception e) {
  stdTTPsetSystemText("ERROR: Failed to start server for receiving " +
            "Storage Commitment results:" + e.getMessage() + "\r\n");
  return false;
}
try {
  t1 = System.currentTimeMillis();
  test = true; i = 0;
  while ( ( test ) && ( i < 10 ) ) {
     test = !(dcmOpen( dcm_send ));
     if ( test ) {
    stdTTPsetSystemText("dcmOpen: retry @ " + Boolean.toString(test) +
        " && i= " + Integer.toString(i) + "\r\n");
     }
  }
  t2 = System.currentTimeMillis();
 stdTTPsetSystemText( "Connected to " + strAetIP[0] + " in "
                   + ((t2 - t1) / 1000F) + "s\r\n");
  t1 = System.currentTimeMillis();
  dcm_send.send();
  t2 = System.currentTimeMillis();
       dcm_send.close();
stdTTPsetSystemText("Released connection to " + strAetIP[0] + "\r\n");
} finally {
  dcm_send.stop();
}
stdTTPsetSystemText("dcm_send successful" + ";\r\n");
```

4.3 Implementation of RISPad-Srv and Clt

The RISPad-Srv was designed as a http-server also using the Socket.IO engine 1.0 for real time bi-directional event driven communication with the clients [43]. Setup of an http-server is quick and easy to achieve with the Node.js® environment (**Listing 7**).

Listing 7: *Easy configuration of the RISPad-Srv using Node.js®*

```
//start the server
//.server(app) - solution
http.listen(9080, function() {
    console.log('node-server connected to: 9080');
});
```

Once the server is running the RISPad-Clt is accessible using a standard browser via the network. RISPad-Clt implements a simple intuitive GUI that structures working with the application into (1) authentication (**Figure *14***), (2) adopting the HR-DCD as web-cam, (3) image acquisition.

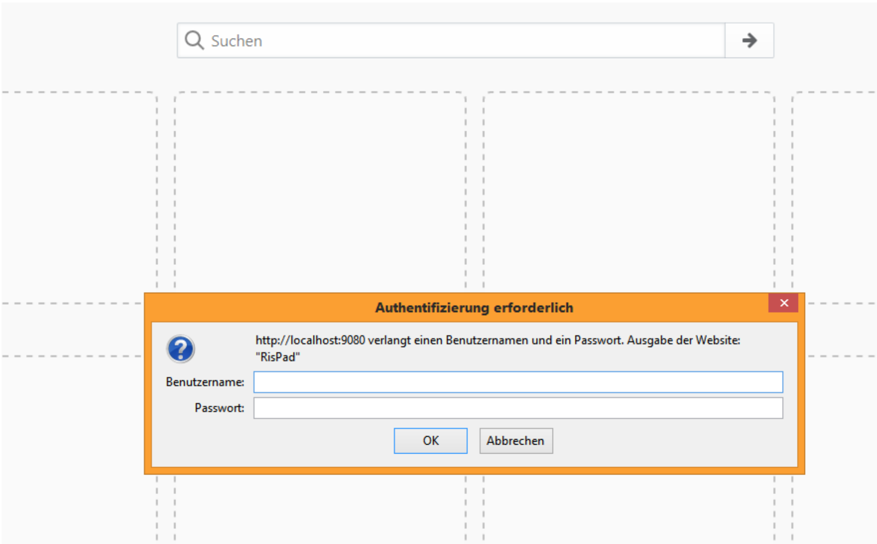

Figure 14: *Authentication using the RISPad-Client*

At the beginning of each RISPad-Clt session the browser requests permission to use the HR-DCD, which is supported as web-cam (**Figure *15***).

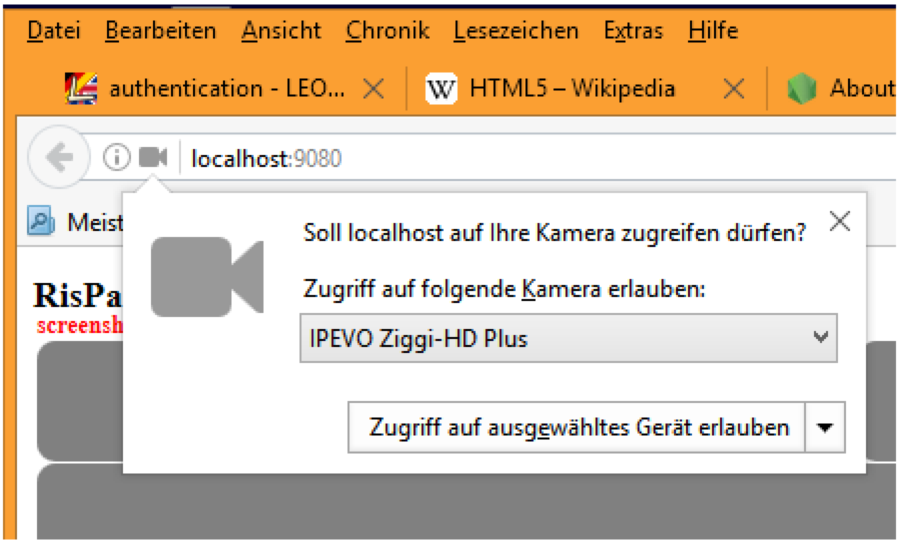

Figure 15: *Permission request to use the webcam, which corresponds to the proposed HR-DCD.*

After access to the HR-DCD is granted images of evidence documents etc. can be acquired easily via the GUI and linked to an appropriate examination. The image acquisition using the GUI was built in a way to induce a work-up cycle starting either with taking a screen shot from the video screen in the right upper corner of the GUI or with choosing a patient's entry at left lower corner. In a movement that leads the user anti-clock wise one time around the interface the complete documentation is done. After taking a screen shot by clicking on the video screen, this screen shot appears in the large viewing control window, where focusation of the image can be checked. In case of an unfocused capture, simply clicking the video screen again replaces the old screen shot with a new one. At the upper margin of the RISPad-Clt a thumb nail ribbon applicable for 12 images is displayed. Clicking one of the thumb nails brings the displayed respective image back onto the viewing control window, where they can be viewed or deleted. In the left lower corner a table displays all the data of the available examinations, where the images can be linked to. For rapid access to an examination notification file a query- and a refresh-function is implemented. Finally the acquired images are stored at the RISPad-Srv and sent to the PACS after conversion to RGB-files, when the 'link to dicom'-button is pressed (Figure 16). The RISPad-Clt passes a so-called tie-structure to the RISPad-Srv, which resembles a directory structure named with a unique tie identifier that avoids double work-up of RISPad-documented files. Additionally, the original notification file is included into the tie to assure correct linkage of the RISPad-documentation to the proper examination in the PACS, where the DICOM-entries are used to find the correct study. Thus, the screen captures could be flagged with the respective DICOM UIDs of the related examination in order to instruct the PACS how to relate the newly entered clinical records to the image data, where representation of the RISPad images as a series in the corresponding study was implemented.

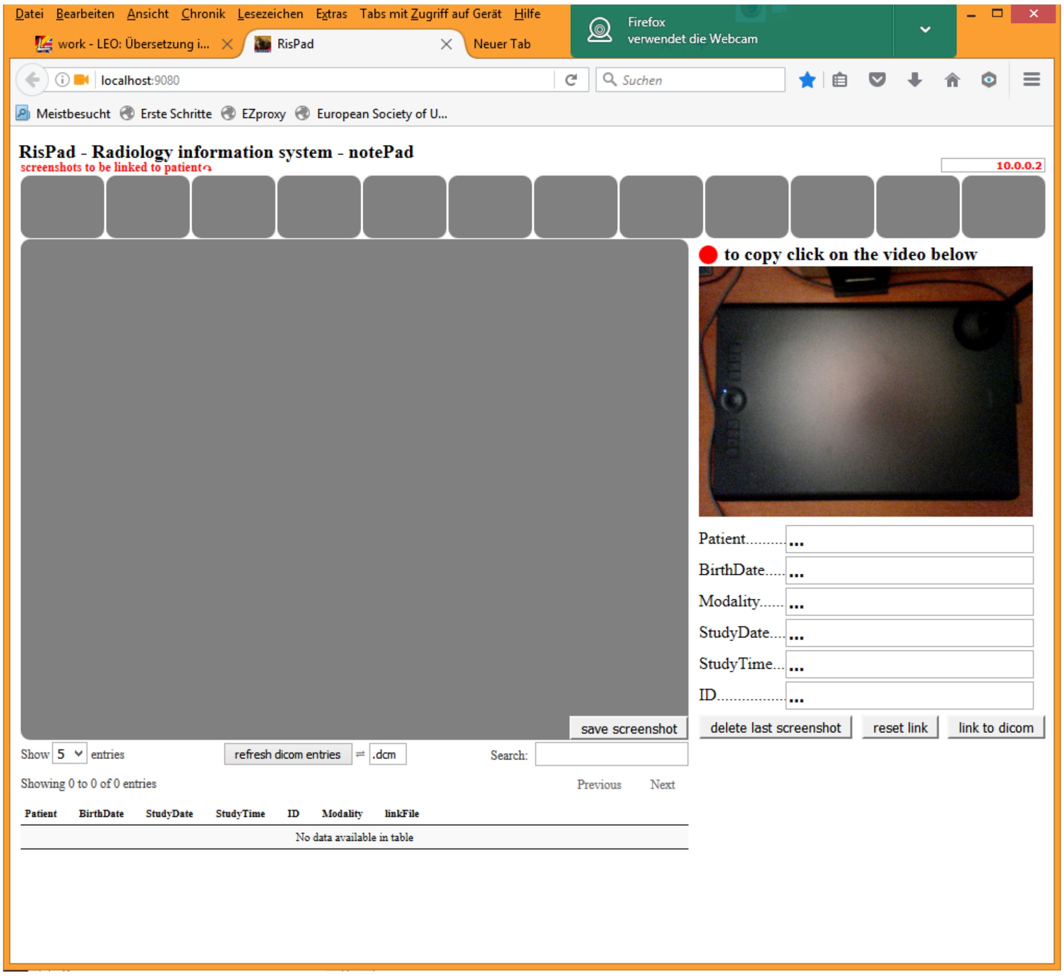

Figure 16: *The GUI of the RISPad-Clt was clearly arranged. The work-flow leads the user anti-clockwise one-time around the working place.*

4.4 Testing RISPad

As a proof of concept beta-testing was performed at the department of Radiology at the University Hospital Tulln. The software was implemented in October 2016 and tested since then for about 7 months. In total only few (n= 6) crashes of the server were noticed, which were related to break downs of the regional network connections or an over-load of available notification files. Latter situation occurred because the auto-delete function implemented in the software was deactivated during the test phase in order to preserve all data for later evaluations and comparisons. Auto-deleting using RISPad is usually performed when the transmission of a RISPad-series is echoed successful from the PACS to the RISPad-BkS-SCU. Simply restarting RISPad solved the problem.

The possibility to run RISPad in a standard web-browser was very much appreciated by the in-house IT-team. So far more than 100 documentations were finished successfully. An elaborated assessment of the RISPad real world application was performed 2017 by Ganster B., co-author of the RISPad-project, who will report the results from the assessment in a related thesis. Typical scenarios for image documentations using RISPad were: (1) biological material extracted during an intervention (compare **Figure 12** and **Figure 13**), (2) documentation sheets of interventionally used devices and (3) informed consent sheets modified on-site due to newly added information from the patients.

All personal data used to test the RISPad implementation was accessed according to the Austrian Health Telematic Law 2012 – GtelG 2012 [44].

4.5 Cost Effectiveness

A total work load of app. 160 working hours of coding all RISPad-components was noticed. The costs of one coding hour was calculated with 85,-- EUR [45]. Asset costs for the HR-DCD were 120,-- EUR including shipping costs. Since all other means for the development of RISPad were free of charge. The total costs of the RISPad-project development were 13 720,-- EUR.

The costs for the proposed server hardware were estimated at 1 500,-- - 2 000,-- EUR, while the costs for the hardware needed for the RISPad-WkP were judged in a range between 300,-- and 600,-- EUR.

Since RISPad works like a copy machine and does not produce any new health data, no costs for registration of RISPad as a medical product were calculated. With a total of app. 18 000,-- EUR for the complete implementation to reach the aimed PoC the RISPad resides in the absolute low-cost segment of HIT developments.

5 Discussion

RISPad implements a radiological DHC-application dedicated to the direct integration of clinical and medico-legal VL-documented data into correlated modality-specific radiological studies stored in a PACS.

5.1 Documentation of Radiological Information

DHC is engaged with finding innovative technological solutions to integrate newly available technologies into the routine work flow of health care professionals, thereby improving the quality of patient management and medical documentation. In this context digital photographic documentation and analysis of events, facts or situations has become a natural part of many health care task, which was acknowledged, e.g., by definitions as IHE-WIC profile [12]. In radiology image based documentation of diagnostic examinations plays a central role, but this relates to modality specific diagnostic imaging only. Radiology, per se, widely lacks systems enabling the documentation of relevant evidence documents or photographic systems documenting biological materials collected in radiological interventions, although few reports can be found in literature [35]. This is *a fortiori* of relevance, since it could be shown that even simply providing some technical information about the diagnostic imaging during radiological reporting significantly improves the report quality [25]. Using RISPad now any kind of capture considered relevant for a radiological examination can be documented together with its corresponding examination as shown by the presented PoC.

Moreover, the missing documentation of clinical evidence documents can cause medico-legal and often expensive consequences also in radiology [7], which implies a certain need to be able to document clinical incidents, like e.g., a possible pregnancy, observed allergenic reactions or special complications explained to a patient, most suitably on site. Therefore, RISPad offers on-site documentation and image capturing, which was an important claim during the prototyping process. Besides the documentation of clinical evidence sheets, documents or forms, the direct depiction of biological material collected from an intervention is also of interest, as radiology is science in progress. For instance, in the department of radiology, where the PoC that RISPad principally works was evaluated, thrombectomy in intracranial vessels is performed. Though this procedure is potentially life-threatening, it is the only intervention possible in severe cases of acute ischemic stroke that can save the life and health of a patient. Clearly, this is a field of intensive research, which is now supported by the RISPad-application that allows documenting not only the angiographical images of this procedure, but also the material extracted from the vessels (**Figure 17**).

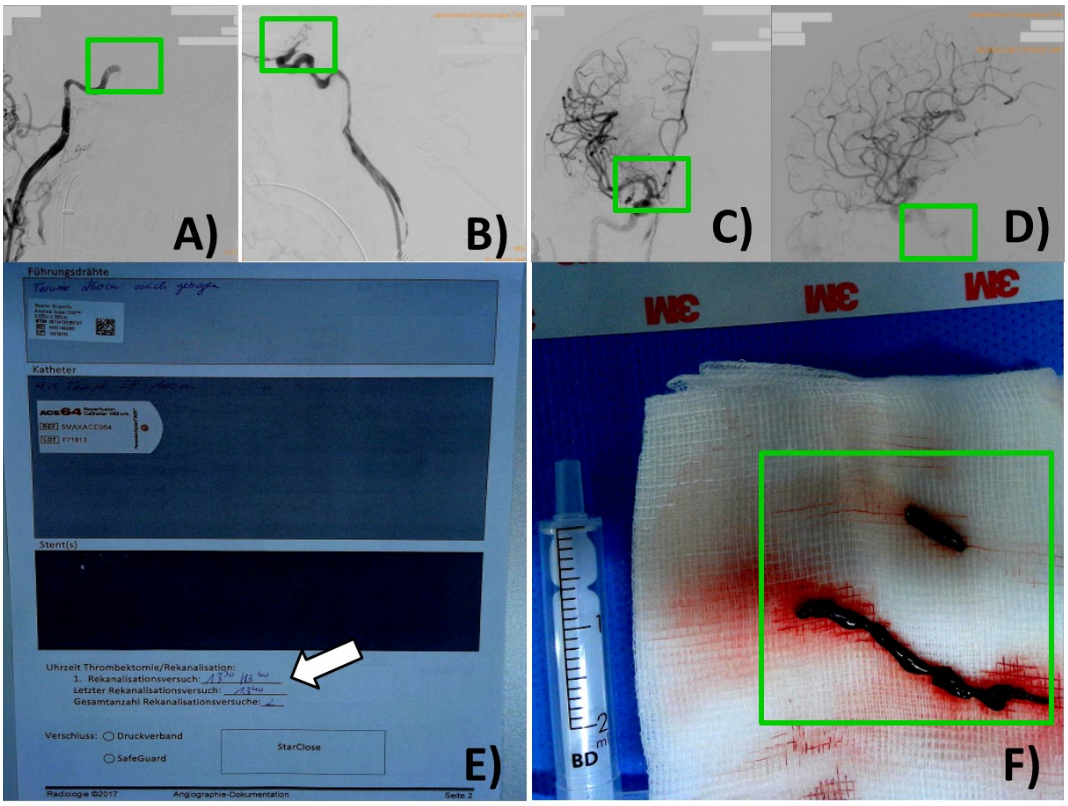

Figure 17: *Example of a complete radiological and clinical documentation using RISPad. All images are stored together in a single study in the PACS. First row: The angiographic images depict the region (green rectangle) where a thrombus was extracted from a major brain artery. Images A & B demonstrate the occlusion. Images C & D show complete filling of all vessel branches after recanalization.*
Second row: A copy of the intervention protocol notifying about the exact recanalization time (white arrow) is provided and is, therefore, available when the report is dictated (image E). The extracted thrombotic material is documented for further analysis (image F).

In this context RISPad could be shown to perfectly document the thrombus material extracted from the brain in high resolution images (**Figure 13**), which allow judgment of the thrombus composition. Latter fact was found relevant for therapeutical considerations in this type of treatment [8].

5.2 Project Planning

The implementation RISPad was aligned to the IHE-recommended four step standard process and followed the proposed standards to maintain a maximum of interoperability with existing HIT systems, especially the PACS. However, the recognition of recommended standards by the IHE initiative is an important step to allow the rational exchange of data between different health information systems, but it should be noticed that this was and still is a process also driven by industrial vendors and developers and not only by HC-professionals. For instance, the consultation of clinical experts is mentioned only in one of four steps in the standard IHE-process for the software development by the IHE-initiative Europe [11]. This underlines the important role of the interdisciplinary DHC-skills, which were useful in all phases of the implementation of RISPad yet to find the most appropriate IT-solution suitable for frictionless working in the daily radiological routine.

The use cases defined in the first step of the IHE-process were all based on the assessment of radiological routine work flow and apply therefore to realistic assumptions concerning a rational documentation in clinical quality management. During the second step a rather complex structure of the intended documentation system became evident in prototyping, which was overcome by modeling the process as close as possible to routine work flow. In the third step the designed prototype was implemented, where a high value was set on cost effectiveness. This was achieved using widely accepted, well tested free ware tools and development kits. Frictionless function of the RISPad application prototype in beta-testing brought a positive proof of concept in a real world environment in a radiological department and at the date of the publication of this thesis RISPad was stable running in a 'real world' - scenario for already several months with only few incidents due to network problems. As over time RISPad changed to a real world application, a complete documentation and assessment of the RISPad software could be performed in the last step of the IHE-recommended process, which is reported separately in another thesis dealing with RISPad. In every step of the development process RISPad satisfied all demands of conformity required for a light weight, fast accessible low-budget documentation tool for daily use in the routine radiology work flow.

5.3 Relevance of CUCs

Preceding prototyping RISPad real world scenarios were identified during the project analysis. In one case an informed consent sheet of a radiological intervention vanished from documentation needed later in a law-suit. Informed consent sheets are currently documented in the patient record at the ward, but since many patients claim questions directly at the radiological department often patient records and report forms get separated. There is no possibility for the radiologist to check, whether a report form is correctly reentered into the patient record after sending it back to the ward. Using RISPad now the informed consent sheet inclusively eventually written amendments can be directly documented within the respective DICOM study in the PACS.

Another real world scenario revealed that an allergenic reaction after administration of contrast media intravenously was mentioned on an informed consent sheet from an earlier radiological examination two years ago. This sheet was part of an old patient record that was not available at the time of the current examination. There was no way to find out about this fact, because the old sheet was not even stored electronically, thereby also rendering general health record collections like the Austrian ELGA-project insufficient [32]. On the other hand, using RISPad for an immediate documentation of the informed consent sheet together with the DICOM-study in the PACS would have solved the problem and, therefore, would have increased the patient's safety, since the old study itself was very well available in the PACS. The same is true, for instance, for the documentation of a MR-compatible device implanted in a patient's body. Simply documenting the implant's ID-card with the corresponding DICOM-study allows the fast access of these data in the PACS, when the patient comes a second time for a MR-examination.

5.4 Structure, Prototyping and Testing RISPad

RISPad is an easy-to-use tool for the documentation of clinical evidence documents or captures from biological material obtained from interventional procedures. Established means of a radiological administration widely lack the possibility to generate and assess any documentation files where they are actually needed. From **Figure 9** it is evident that the radiologists and RTs working with the RIS, the PACS, the connected modalities and the image viewers cannot directly access information's from the HIS. All clinical relevant information, e.g., adverse reactions etc., are (or should be) reported in the HIS. Since HIS-clients are not supported by most of the available image viewers, where the radiological report is generated, potentially relevant information stored in the HIS is not easy to integrate into the radiological work flow. Thus, in the process of designing RISPad it became evident that working efficiently with various HIT-facilities in radiology does frequently not fail because the requested data is not available, it most often fails simply because of where the data is available. Since passing of information within the various HIT-systems is mainly directed bottom-up, it is also evident that the best place to acquire the radiological evidence sheets or captures is on-site at the department and the best place to store them is the PACS. Therefore, RISPad captures document images on-site and stores them directly in the PACS. During beta-testing the PoC showed that this was indeed the most practical way to integrate a documentation system like RISPad in the radiological work flow.

On-site documentation using RISPad was instantiated by designing RISPad as a server-client network solution. So, the user is able to access the application via a standard web browser on a light weight PC with a connected HR-DCD. Since the HR-DCD is recognized as webcam and the application runs inside the browser no installation of additional software on PCs in a radiological department necessary, a fact very much appreciated by the in house IT-department where the beta-testing was performed. The GUI implemented in the RISPad-Clt is straight forward to capture an image, to control the focus and to link the image to the correct examination in order to best fit the radiological work flow. No additional functionalities are implemented not to entangle the user. Capacity properties of RISPad were exactly adopted to the clinical needs and were never exceeded so far.

A central part of the communication design in the RISPad-project is the use of a notification image instead of a real DICOM-query (**Figure 11**). Assessing the daily work flow revealed that many employees did not cope well with a classical database interface, where queries are usually performed. Sending images from one network-node to another, on the other hand, is permanently done. Therefore, the process of sending images was adopted to RISPad and this met highest acceptance in the beta-testing. No erroneous links of captures to inappropriate studies in the PACS were noticed so far.

The IDE and SDKs used in the development of RISPad provided an excellent performance and allowed parallel coding of parts of RISPad in Java®, in JavaScript® and HTML5® within the same environment. Additionally, the NPM of the used Node.js® server was easy to handle using the respective plug-ins provided by the IDE. Allover, the implementation of the project was possible straight forward, because there was the opportunity to use and recombine many detail solutions from libraries provided for most of the used components. Since this remarkably reduced testing of all subfunctions, the work load to develop RISPad was rather small. The proposed prototype was implemented in a real world-scenario, where RISPAd ran stable so far. Due to the freely available software components used to develop RISPad and the high performance HR-DCD with the ability to store images of printed text, which can be even used for automatic text translations.

RISPad was designed as a light weight, low-budget tool with high cost-effectiveness. Since the total work-load was kept to a minimum by using freely available standard tools, the calculated costs for the development of RISPad are extremely low. The costs for the used HR-DCD are also low compared to other available products. Since the application runs in a standard browser, where a 64 bit operating system with 64 bit browser software is recommended, the requirements of the RISPad- Clt are also minimal.

5.5 Limitations

The major limitation in estimating intercooperatbility of RISPad with established HIT-systems arises from the fact that the software could only be tested in one single department. This is owed to the fact that the members of the RISPad-project team had only limited time for developing and testing RISPad available. Furthermore, empirically the dialog with other IT-departments than the own in-house one is difficult, so that RISPad was installed at the UK Tulln, department of radiology. However, unambiguously a positive PoC was found using RISPad.

Furthermore, yet the number of stored clinical evidence documents is still rather small (about 150 cases). Results from longer observation periods are pending, which will be assessed continuously.

Finally, RISPad-Srv was implemented using Node.js@ as server side runtime. Hence, the survival of the RISPad-project could rely on the persistent support of the sponsors of the Node.js® project.

6 Conclusion

The documentation of clinical data collected during examinations in radiology using RISPad, a light weight, low-budget server-client documentation system, was shown to be feasible. Evidently, the development and implementation of assistive DHC tools able to add clinical data as captures to radiological examinations in the PACS is of importance, as some of demonstrated use cases even had a medico-legal law-suit background. This emphasizes the need of flexible documentation systems in radiology, which could potentially further increase standards in quality management in radiology. Though few limitations concerning RISPad were found, a positive PoC for RISPad could be presented in this thesis, which is groundbreaking for even more elaborated real world applications.

Literature

[1] A. M. Cormack, "Representations of a function by its line integrals, with some radiological applications.," *Journal of Applied Physics,* vol. 34, pp. 2722-2717, 1963.

[2] A. M. Cormack, "Representations of a function by its line integrals, with some radiological applications, II.," *Journal of Applied Physics,* vol. 35, pp. 2908-2913, 1964.

[3] G. N. Houndsfield, "Computerized transverse axial scanning (tomography): Part I. Description of system," *British Journal of Radiology,* vol. 46, pp. 1016-1022, 1973.

[4] K. Foord, "PACS: the second time around," *Eur J Radiol,* vol. 32, pp. 96-100, Nov 1999.

[5] J. C. Honeyman, "Information systems integration in radiology," *J Digit Imaging,* vol. 12, pp. 218-22, May 1999.

[6] DICOM Standards Committee Working Group 13. (2015, Sept. 17th). *Digital Imaging and Communications in Medicine (DICOM). SUPPLEMENT 15: Visible Light Image for Endoscopy, Microscopy, and Photography (Final Text 1999-07-02 ed.)*. Available: ftp://medical.nema.org/medical/dicom/final/sup15_ft.pdf

[7] Patientenanwaltschaft Vorarlberg, "Jahresbericht," ed: Patientenanwaltschaft Vorarlberg, 2009, p. 26.

[8] D. S. Liebeskind*, et al.*, "CT and MRI Early Vessel Signs Reflect Clot Composition in Acute Stroke," *Stroke,* vol. 42, pp. 1237-1243, 2011.

[9] S. S. Boochever, "HIS/RIS/PACS integration: getting to the gold standard," *Radiol Manage,* vol. 26, pp. 16-24; quiz 25-7, May-Jun 2004.

[10] IHE Europe. (2017, Mar. 18th). *About Us | IHE Europe*. Available: https://www.ihe-europe.net/about-us

[11] IHE Europe. (2017, Mar. 18th). *IHE Process | IHE Europe*. Available: https://www.ihe-europe.net/specifications/ihe-process

[12] IHE Radiology Technical Committee, "Web-based Image Capture (WIC)," in *Integrating the Healthcare Enterprise IHE Radiology Technical Framework Supplement.* . vol. Rev. 1.2 - Trial Implementation, ed, 2016.09.09.

[13] D. W. Bidgood*, et al.*, "Understanding and Using DICOM, the Data Interchange Standard for Biomedical Imaging.," *Journal of the American Medical Informatics Association,* vol. 4, pp. 199–212, 1997.

[14] Wikipedia. (2017, MAR. 22nd). *Digital Imaging and Communications in Medicine*. Available: https://de.wikipedia.org/wiki/Digital_Imaging_and_Communications_in_Medicine

[15] NEMA. (2017, Mar. 26th). *The DICOM Standard*. Available: http://dicom.nema.org/standard.html

[16] NEMA. (2017, Mar. 26th). *C.2.2 Patient Identification Module*. Available: http://dicom.nema.org/medical/Dicom/2016b/output/chtml/part03/sect_C.2.2.html

[17] NEMA. (2017, Mar. 26th). *F.2 DICOM JSON Model*. Available: http://dicom.nema.org/dicom/2013/output/chtml/part18/sect_F.2.html

[18] NEMA. (2017, Mar. 26th). *DICOM PS3.6 2017a - Data Dictionary*. Available: http://dicom.nema.org/medical/dicom/current/output/pdf/part06.pdf
[19] D. Brunold. (2017). *C.7.6.3.1.2 Photometric Interpretation*. Available: https://www.dabsoft.ch/dicom/3/C.7.6.3.1.2/
[20] M. E. Latoschik. (2006, Mar. 30th). *Realtime 3D Computer Graphics / Virtual Reality*. Available: https://www.techfak.uni-bielefeld.de/.../8a.RT3DCGVR-color.pdf
[21] Health Level Seven International. (2017, Mar. 30th). *HL7 Standards Product Brief - CDA® Release 2*. Available: http://www.hl7.org/implement/standards/product_brief.cfm?product_id=7
[22] IHE Europe. (2017, Mar. 30th). *Cross-Enterprise Document Sharing*. Available: http://wiki.ihe.net/index.php/Cross-Enterprise_Document_Sharing
[23] M. J. Franczak*, et al.*, "In emergency departments, radiologists' access to EHRs may influence interpretations and medical management," *Health Aff (Millwood),* vol. 33, pp. 800-6, May 2014.
[24] European Medicines Agency. (2006, Sep. 30th). *ICH Topic E 6 (R1) - Guideline for Good Clinical Practice*. Available: http://www.edctp.org/fileadmin/documents/EMEA_ICH-GCP_Guidelines_July_2002.pdf
[25] H. H. Abujudeh*, et al.*, "Automatically inserted technical details improve radiology report accuracy," *J Am Coll Radiol,* vol. 8, pp. 635-7, Sep 2011.
[26] J. H. Buhk and M. Fleischer, "Radiologie im Verbund der Klinikkommunikation – Herausforderungen, Lösungen und Fallstricke," *Der Radiologe,* vol. 54, pp. 9-18, 2014.
[27] D. B. Larson*, et al.*, "Communication in Diagnostic Radiology: Meeting the Challenges of Complexity," *American Journal of Roentgenology,* vol. 203, pp. 957-964, 2014.
[28] G. Gassert*, et al.*, "Interventional Radiology Workflow Management in the Electronic Medical Record," *Journal of Digital Imaging,* vol. 27, pp. 314-320, 2014.
[29] J. W. Nance Jr*, et al.*, "The Future of the Radiology Information System," *American Journal of Roentgenology,* vol. 200, pp. 1064-1070, 2013.
[30] D. L. Weiss*, et al.*, "Radiology reporting: a closed-loop cycle from order entry to results communication," *J Am Coll Radiol,* vol. 11, pp. 1226-37, Dec 2014.
[31] K. W. McEnery, "Coordinating patient care within radiology and across the enterprise," *J Am Coll Radiol,* vol. 11, pp. 1217-25, Dec 2014.
[32] ELGA GmbH. (2015, Mar. 18th). *ELGA: Technischer Aufbau im Überblick*. Available: http://www.elga.gv.at/technischer-hintergrund/technischer-aufbau-im-ueberblick/index.html
[33] R. J. Cruz-Correia*, et al.*, "Reviewing the integration of patient data: how systems are evolving in practice to meet patient needs," *BMC Medical Informatics and Decision Making,* vol. 7, p. 14, 2007.
[34] P. Coorevits*, et al.*, "Electronic health records: new opportunities for clinical research," *Journal of Internal Medicine,* vol. 274, pp. 547-560, 2013.
[35] S. Ramamurthy*, et al.*, "Integrating Patient Digital Photographs with Medical Imaging Examinations," *Journal of Digital Imaging,* vol. 26, pp. 875-885, 2013.
[36] Node.js. (2016, 16th Mar). *Node.js a JavaScript built runtime*. Available: https://nodejs.org/en/

[37] SpryMedia Ltd. (2016, 3rd May). *DataTables Table plug-in for jQuery*. Available: https://datatables.net/

[38] NPM. (2016, 20th Jan). *NPM - Nightly Patch Machine*. Available: node-webcam https://www.npmjs.com/package/node-webcam

[39] NPM - Numeric Production Mechanism. (2016, 2nd Jun). *Daikon a pure JavaScript DICOM reader*. Available: https://www.npmjs.com/package/daikon

[40] Zeilinger Gunter. (2016, 19th Mar). *dcm4che3*. Available: https://github.com/dcm4che/dcm4che

[41] NetBeans IDE (supported by ORACLE). (2016, Feb. 10th). *NetBeans IDE - Overview*. Available: https://netbeans.org/features/index.html

[42] NetBeans IDE (supported by Oracle). (2016, 16th Feb). *Welcome to the NetBeans Plugin Portal*. Available: http://plugins.netbeans.org/PluginPortal/

[43] The Socket.IO Contributors. (2016, 15th May). *Socket.IO*. Available: https://socket.io/

[44] Bundeskanzleramt - Rechtsinformationssystem. (2017, May 2nd). *Bundesgesetz betreffend Datensicherheitsmaßnahmen bei der Verwendung elektronischer Gesundheitsdaten (Gesundheitstelematikgesetz 2012 – GTelG 2012)*.

[45] freelancermap. (2017, 10th May). *Welchen Stundenlohn kann man als Freiberufler verlangen?* Available: https://www.freelancermap.at/freelancer-ratgeber/10718-welchen-stundenlohn-kann-man-als-freiberufler-verlangen-

Abbreviations

AJAX .... Asynchronous JavaScript And XML [→XML]
CSS .... Cascading Style Sheets
CT .... Computer Assisted Tomography or Computertomography
CUC .... Clinical Use Case
DCD .... Document Camera Device
DICOM .... Digital Imaging and Communications in Medicine
DIMSE .... DICOM Service Element
DHC .... Digital HealthCare
DSA .... Digital Subtraction Angiography
ED .... Evidence Document
EDWC .... ED [→ED] web-client
EDWS .... ED [→ED] web-server
E-Health .... Electronic-Health
EHR .... Electronic Health Record
ELGA .... Elektronische Gesundheitsakte
GUI .... Graphic User Interface
HC .... Health Care
HR .... High Resolution
HIS .... Hospital Information system
HIT .... health-IT [→IT]
HL 7 .... Health Level Seven
HL 7 CDA .... HL 7 [→HL 7] Clinical Document Architecture
IDE .... Integrated Development Environment
IHE .... Integrating the Health Enterprise
IOD .... Information Object Definition
IT .... Information Technology
JPEG .... Joint Photographic Experts Group
JSON .... Java Script Object Notation
MR .... Magnetic Resonance
MRI .... MR [→MR] Imaging

MRT..................... MR [→MR] Tomography
NI......................... notification image
PACS................... Picture Archive and Communication System
PC .... Personal Computer
PoC .... Proof of Concept
RAM .... random access memory
RT .... Radiology Technologist
RIS .... Radiology Information System
RISPad .... Radiology Information Storage Notepad
RISPad-BkS .... RISPad [→RISPad] - background services
RISPad-Clt........... RISPad [→RISPad] - client
RISPad-Srv.......... RISPad [→RISPad] - server
RISPad-WkP .... RISPad [→RISPad] – working place
RWIM .... Real World Information Model
SDK..................... Software Development Kit
TUC..................... technical use case
SCP..................... Service Class Provider
SCU..................... Service Class User
SOP..................... Service-Object Pair
STM..................... Syntax Translation Module
VL........................ Visual Light
UID .... Unique Identifiers
USB..................... universal serial bus
WIC .... Web-based Image Capture
XDS..................... Cross-Enterprise Document Sharing
XML..................... Extensible Markup Language

List of Figures

Listings