IT University
of Göteborg

CHALMERS | GÖTEBORG UNIVERSITY

Achieving Safe DICOM Software in Medical Devices

Kevin Debruyn

CHALMERS | UNIVERSITY OF GOTHENBURG

Student

Kevin Debruyn (820717-2795)

Contact Information

Phone: +46 73 7418420 / Email: kevindebruyn@gmail.com

Course Supervisor

Karin Wagner

Course Coordinator

Kari Wahll

Start and End Date

21st of February 2008 to 30th of March 2009

Size

30 Credits

Subject

Achieving Safe DICOM Software in Medical Devices

Overview

The present document constitutes the project report that introduces, develops and presents the results of the thesis carried out by a master student of the IT University in Göteborg, Software Engineering & Management during Spring 2008 through Spring 2009 at *Micropos Medical AB*.

Summary

This paper reports on an investigation on how to produce a reliable software component to extract critical information from DICOM files. The component shall manipulate safety-critical medical information, i.e. patient demographics and data specific to radiotherapy treatments including radiation target volumes and doses intensity. Handling such sensitive data can potentially lead to medical errors, and threaten the health of patients. Hence, guaranteeing reliability and safety is an essential part of the development process.

Solutions for developing the component from scratch or reusing all or parts of existing systems and libraries will be evaluated and compared. The resulting component will be tested to verify that it satisfies its reliability requirements. Subsequently, the component is to be integrated within an innovating radiotherapy positioning system developed by a Swedish start-up, *Micropos*.

While remaining objective, I will examine how the outcomes of my practical cases converge with what is discovered in the literature: a massive lack of conformance to DICOM exists among medical systems that produce DICOM files. Endeavors by the manufacturers developing medical devices are still required to universalize the interoperability proclaimed by the creators of the standard. Despite that imperfect context, a reliable software component was achieved and is presented in this report.

Keywords

DICOM, Medical Device, Safety-Critical Systems, Software Standards

TABLE OF CONTENT

INTRODUCTION

Context

The prostate is an organ present in the male mammalian reproductive system and is involved in the production and expulsion of semen. Prostate cancer is the most common type of cancer for men, most prominent among male subjects over fifty years old. The statistics for death tolls show that casualties related to prostate cancer have tripled in the last 50 years in Sweden (1).

There are several medical treatments for prostate cancer, all of which are concerned with the suppression of the cancerous tissues, such as surgery and internal or external Radiation Therapy (RT). Balancing the probability of benefits brought by the ablation of the tumor against its risks of negative ramifications for the patient's health condition is a compromise that must be carefully considered by clinical oncologists (2)(3).

In effect, due to the proximity of surrounding non-cancerous organs, i.e. the rectum, bladder and the prostate itself (4), unavoidable damage is dealt to healthy tissues. Therefore, the development of medical devices with stringent safety requirement comprises exhaustive risk management processes including risk analysis and the implementation of risk control measures.

Despite those drawbacks, radiology and oncology are disciplines that have known great advancement thanks to the introduction of information technology in medical processes. They have greatly benefited from the evolution of imaging techniques in medicine, and improved the manner information can be accessed for educational or diagnostic purposes (5).

External radiotherapy involves the treatment of cancerous cells by beam radiation emitted from a machine called linear particle accelerator as opposed to internal radiation therapy or brachytherapy, where radiations are emitted from inside the body. Of course, the accurate placement of the patient on the table during exposition to the rays is of paramount importance. Ideally, the exact location of the tumor, defined as the patient isocenter, matches the radiation focal point.

This is, however, hardly possible to achieve in practice, partly because of the exact localization of the tumor inside the body is difficult to pinpoint. Plus, daily setup errors can occur from session to session and even within sessions, due to random organs movement such as by coughs or varying bladder size and so forth.

Medical imaging like computed tomography (CT) or X-ray scans are used in modern radiotherapy to capture the internal patient anatomy before treatment. It allows radiologists to determine the positions of body organs and tumors.

This is used to establish a diagnostic with the help of a radiation treatment planning system (RTPS). Subsequently, the patient is placed during treatment according to that information, using body markings on the skin and laser pointers positioning before every beaming session.

To cope with this uncertainty, a number of broader areas, or clinical safety margins, must be outlined during treatment planning around the central target. It is crucial to ensure that all the cancerous tissues as well as vicinal secondary infections (subclinical diseases) are exposed to radiations. Hence, there is an accepted threshold for radiation risk to healthy tissues.

The tumor area to radiate is called Gross Tumor Volume (GTV) around which is drawn a Clinical Target Volume (CTV) that comprises secondary infections. The CTV is then surrounded by a Planning Target Volume (PTV) that contains two additional safety margins (the internal margin and the set-up margin). Research is active towards providing new techniques of reducing those margins, as better accuracy will achieve a minimized impact to the organs in the vicinity (2) (Figure 1).
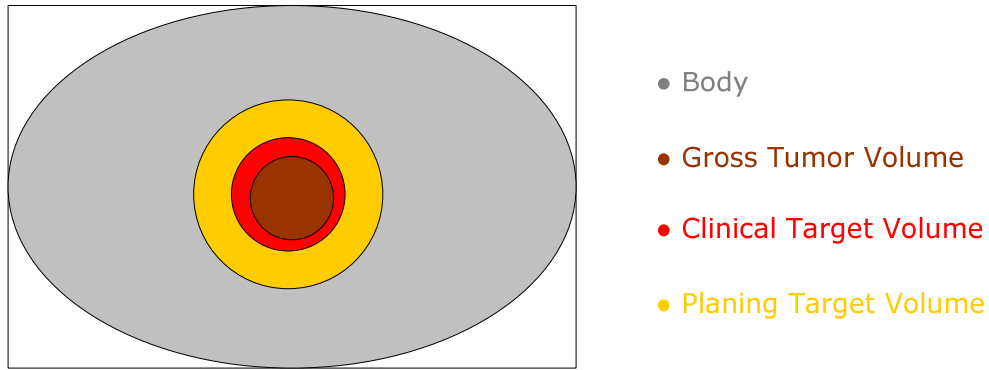
Figure 1: Schematization of a CT scan showing the geometric areas used in radiotherapy

Micropos (6) is a research company part of the Chalmers research centre and innovation business incubator concerned with the real-time tracking and positioning of organs and tumors during radiotherapy. The goal of their *4DRT* (Four Dimensional RadioTherapy) system is to reduce the medical margins around body areas submitted to radiation. This is in order to reduce the risks of complications involved with prostate cancer treatments such as incontinence and impotence.

Micropos' 4DRT system is expected to improve current radiotherapy practices by decreasing the side-effects due to radiations, the clinical pressures in terms of capacity and costs, and the duration of the treatment. These objectives can be attained by achieving a greater precision, hence diminishing the size of the target area and shortening treatment lengths by delivering greater doses at a time. *Micropos* intends to increase the accuracy of RT treatments by tracking the position of the tumor in real-time during beaming sessions using radio transmission.

The standard prostate cancer radiotherapy treatment cycle starts with the acquisition of internal body images, i.e. scanning the body of patient or parts therein. Medical imaging modalities, such as CT-scans, magnetic resonance (MR), x-rays and so on, output a series of 2D images that can be used to generate 3D models of the regions of interests using multi-planar reconstruction (MPR). Radiologists use those images or models to pinpoint tumor areas and draw margins around the target volumes to radiate. Depending on the size and shape of those regions of interests, a treatment dose planning is established to schedule the radiation treatment sessions. During those sessions, the patient is submitted to radiation beams of specific duration, intensities and angles. The overall process is depicted in Figure 2.

Micropos' system is to be integrated within the current treatment workflow in clinical departments or radiological centers, from initial diagnostic through treatment sessions. How *4DRT* fits in this process is done by the surgical implantation of an antenna inside the body of the patient at the beginning of the cycle and its removal in the end. The antenna is implanted inside or nearby the prostate, close to the tumor. During system operation, the antenna is plugged into the table on top of which the patient lays, which is also part of the system and contains radio sensors. The antenna emits radio waves that are received by the table and the information from the sensors is used to calculate the relative position of the tumor to the table with millimeter-wise precision.

Superposing the patient isocenter (the tumor itself) and the linear accelerator focal point is a matter of aligning several coordinates' offsets. The distance between the antenna and the tumor can be revealed by a CT-scan picture once the antenna is inserted inside the patient. The offset between the antenna and the table is acquired via the radio sensors. Finally, the distance between the table and the linear accelerator is calibrated once at installation of the system and during maintenance.

This is where the integration of a DICOM component in the main system could alleviate the workload of the medical staff by automating the process of extracting the location of geometric areas of interest as well as the position of the antenna from DICOM files.
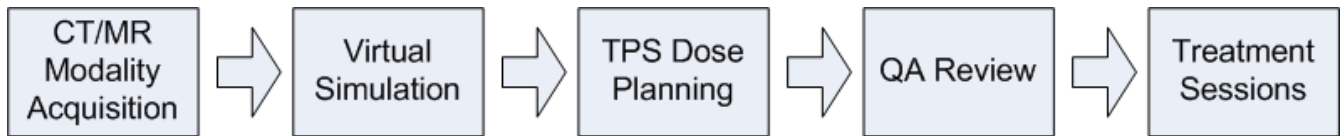
Figure 2: RT standard workflow

DICOM

The diffusion of medical images acquired by means of some modality could induce problematic situations when the identification information associated with the pictures is lost. Separating patient or study information from the images themselves may render orphan data useless. Another problem pertains to buying medical equipment to be integrated in existing systems. The buyer must be certain that the purchased devices will interface and communicate correctly with the structure in place.

In order to prevent such undesired events of information loss and to facilitate the exchange of information between devices from different vendors, medical imaging needed a solution for the acquisition, diffusion and storage of images. Those concerns were embodied in a now internationally adopted de facto standard widely spread and used amongst manufacturers throughout the world: the DICOM standard.

DICOM stands for Digital Imaging and Communications in Medicine (DICOM) and is currently in its third version (DICOM 3.0), based on the previous ACR-NEMA 2.0 standard. It has been created in 1992 by the American College of Radiology (ACR) and the National Electrical Manufacturers Association (NEMA). It is a nonproprietary endeavor meant to enable and foster interoperability amongst different devices' manufacturers, although it does not warrant it (7).

It consists of a vast collection of rules for data structure and encoding, transfer syntax, association establishment, network protocols, messages exchange and file and directory format for storing, querying, retrieving or printing information. It enables the organization of medical information in the form of a centralized object-oriented database called Picture Archiving and Communication System (PACS) accessible from any compatible medical equipment and systems within the host infrastructure, e.g. a hospital network. The information in question typically gathers medical images, clinical information on patients, studies, diagnostics and so forth.

Not only are DICOM files able to store basic 2D pictures, but also series of 2D pictures representing 3D scenes, videos, overlay layers of text or images and satellite textual information such as encoding schemes. As a matter of fact, there is sub-section of DICOM for each modality: X-Ray, CT, RT, MR, etc. What is relevant to this thesis is of course that section of DICOM applied to radiotherapy, also known as DICOM RT. Applied to radiology modality that means that DICOM can support and render information about the Gross Tumor Volume, Clinical Target Volume and Planning Target Volume fields outlined using a Radiotherapy Planning System.

The entire specification of the DICOM standard is over 3500 pages long and divided into a set of 16 separate yet interrelated documents covering the wide range of medical fields involved. Each document is written in an unambiguous formal fashion like an engineering specification and constituted of an invariant part referring to one or several appendixes that can be updated independently, hence guaranteeing the modularity of the ensemble (8). The different parts are continuously corrected and extended in parallel by different working groups, members of the DICOM committee. The whole standard can be downloaded freely from the official NEMA portal (http://dicom.nema.org/) (9).

6
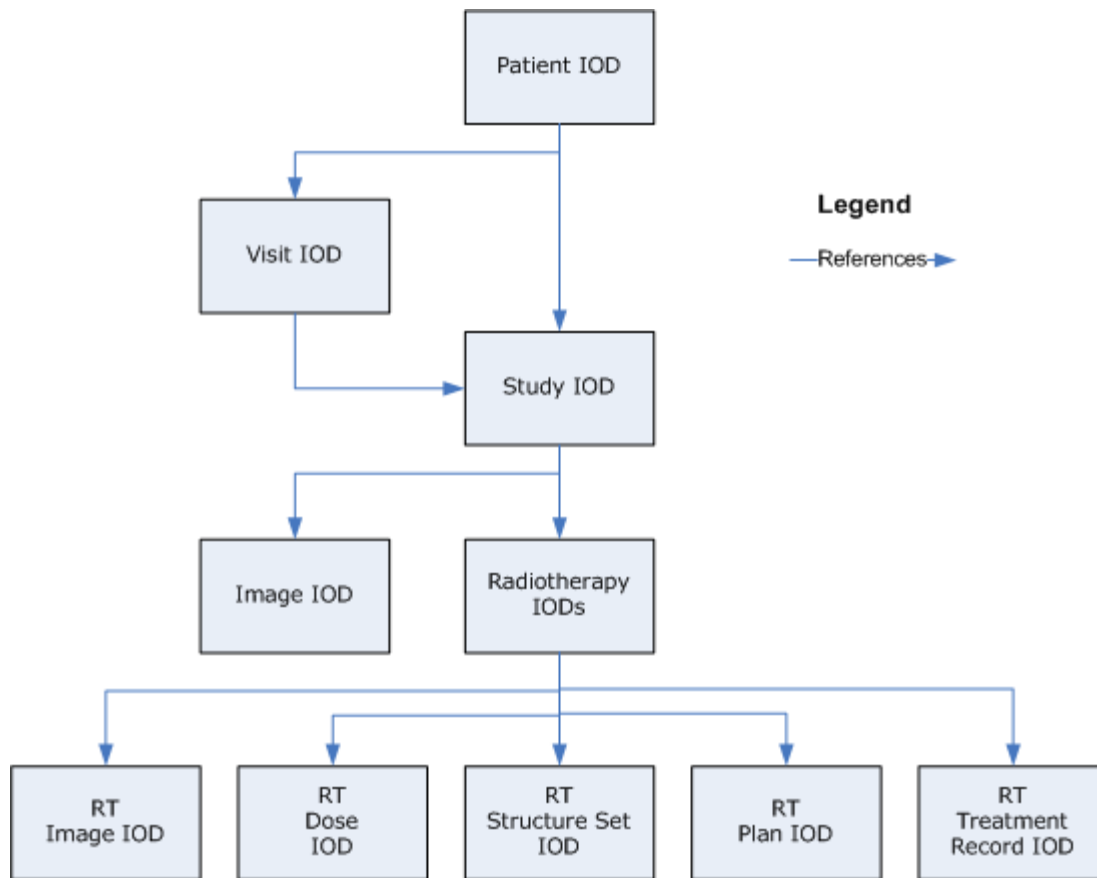
Figure 3: Simplified DICOM Information Object Model (10)

For the sake of brevity and clarity, only the parts pertaining to object modeling, data encoding and file formatting will be synthesized in this introduction, namely parts 3, 5, 6 and 10, although each part relates to others to some extent. In particular, I won't linger on the less relevant networking and message services aspects of DICOM.

Nevertheless, the reader interested in DICOM is encouraged to pursue his/her education on the topic by browsing the literature written on the standard and its underlying concepts (11)(12)(13)(14)(15)(16)(17)(18)(19)(20), on the DICOM Structured Report (21)(22) and on the specific DICOM extensions for radiotherapy (DICOM RT) (23)(24). (Notice that despite its more informal format as presentation slides, (24) was written by an expert DICOM member of the DICOM RT Working Group who thus participated to the development of DICOM). A quick introduction is also provided in the first chapter of this report.

DICOM specifies Information Object Definitions (IODs) that defines the blocks of information persisted in DICOM files representing real-world medical and imaging information objects, for instance a patient, a study, an image of some modality or a dose plan or a treatment record. The Information Objects are designed in an Object-Oriented fashion and organized in a well-defined Entity-Relationship model (see Figure 3).

The IODs are composed of various attributes that can be mandatory, conditional or optional, as well as private (defined by the manufacturers) for additional flexibility. Those attributes describe the properties of real-world objects and are grouped in modules that represent a high-level of semantics. For instance the Patient module comprises the Patient Name, Patient Sex, Patient Birthdate, etc. attributes. The IODs, modules and attributes are all defined in part 3 (PS3.3 (10)) of the standard, a 1000 pages document.
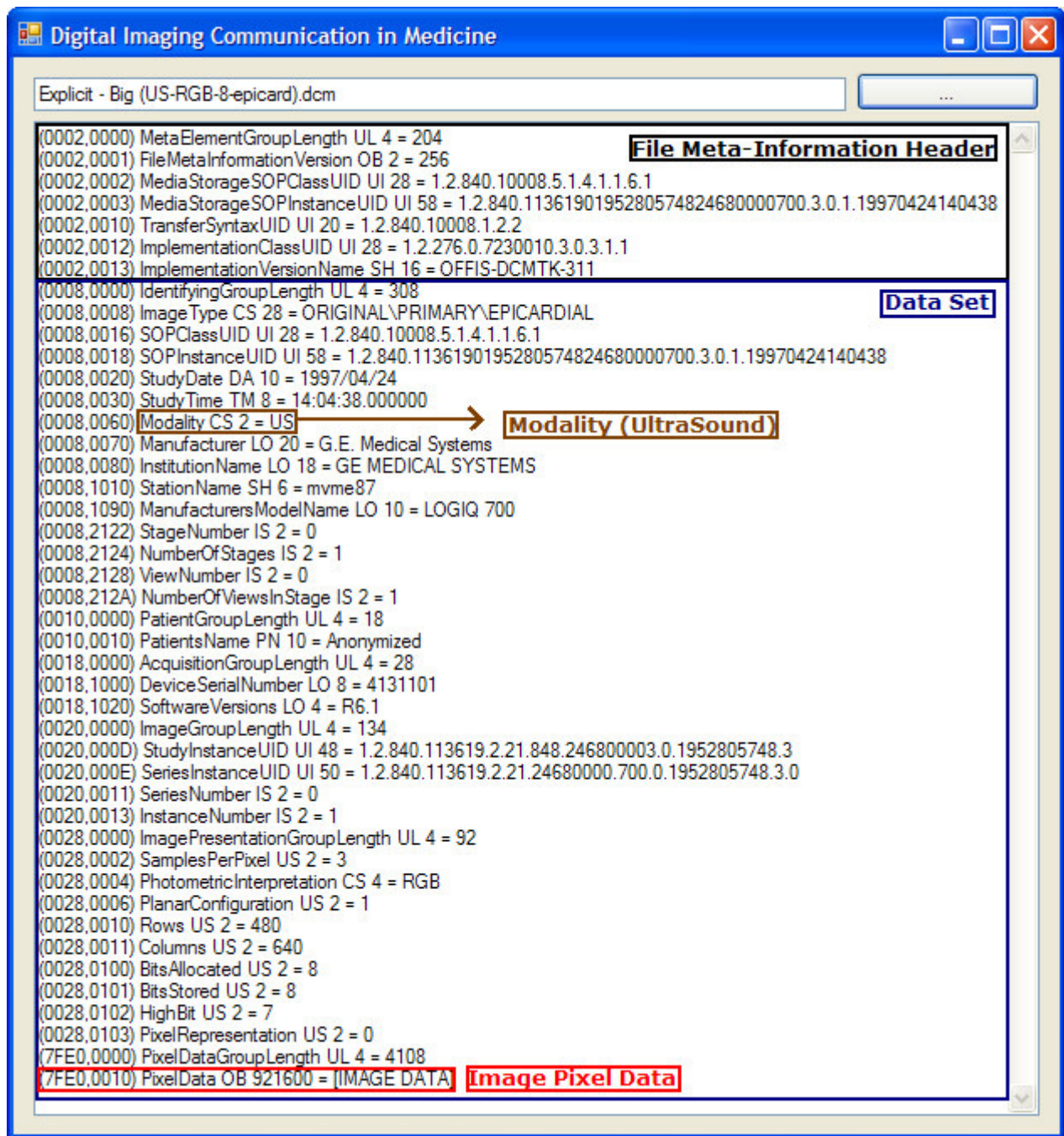
Figure 4: Decoded and Interpreted Content of a DICOM File

Objects in DICOM are defined being either normalized or composite. Normalized Object contain a single logical entity in the DICOM model of the real world; they refer to other Normalized Objects instances to which they are logically connected by UIDs (Unique Identifiers) numbers that are attributed in such manner to be absolutely unique, as in a normalized Entity-Relationship database. Composite Objects on the other hand encapsulate parts of several entities in one IOD such as a RT Image containing Patient information within and are kept in the specification of the standard to guarantee backwards compatibility with previous versions thereof.

Part 5 defines data structures and encoding, as well as the Transfer Syntax (TS). The unit of information in a DICOM file is called Data Element, and represents one attribute after it has been encoded following the rules defined in (25). One Data Element is constituted of a Tag, a Type (also known as VR for Value Representation), a Length and a Value. A set of Data Elements sequentially arranged by their Tag number forms a Data Set. As stated previously, a Data Set encapsulates an Information Object before it is stored in a file or sent across a network.

Part 6 defines a Data Dictionary that is used to retrieve the Element Name, Type and Value Multiplicity from a Data Element Tag number (26).

Part 10 of DICOM, also known as PS 3.10-2008 (27) in its present form, defines *media storage and a file format for media interchange*, in other words contains the specification for DICOM files. A DICOM file is composed of a file meta-information header that contains information about the physical file itself (see Figure 4). That header is followed by a Data Set that represents an Information Object encoded according to the rules and protocols defined in Part 5.

DICOM files are created by compliant medical devices before they are stored, generally in Picture Archiving and Communication Systems (PACS) that are networked databases systems for picture storage and access. The name of the file must have no special signification and should never be utilized to identify the information within. Instead, it is the role of specific Data Elements in the file to uniquely specify the patient or study documented.

Figure 4 unveils the content of a DICOM file after decoding its raw data and using the Data Dictionary to interpret the data elements. Each line represents a Data Element. The numbers between parentheses are the group and element numbers constituting the Tag Number, following by the name of the Data Element, its VR, Length and Value. Note that this screen capture was taken from the first prototype that I developed to acquire some knowledge about the DICOM standard as explained in following chapters. For instance, the Data Element with Tag (0008,0060) in Figure 4 indicates the modality of the image whose bits are stored at the fixed Tag (7FE0,0010) and (0010,0010) is the Patient's name.

DICOM RT is the name given to the set of extensions of DICOM covering the information pertaining to radiotherapy. More precisely, DICOM RT is composed of five IODs subdivided into the modules containing radiotherapy-related attributes. Namely, those Information Objects are RT Image, RT Dose, RT Structure Set, RT Plan and RT Treatment Record (see Figure 3). DICOM Files containing those RT IODs are used through the whole radiotherapy process flow to store information about the patient imaging, diagnostics, dose planning and follow-up treatment.

Problem Area & Problem Statement

Micropos intends to develop a new functionality for its real-time positioning system. The new component shall be able to read certain information relative to radiotherapy from DICOM files and import it in the main system. More precisely, in addition to patient information, the system shall be able to read the position of the tumor and the offset of the antenna from a DICOM file. It will be the task of the radiologists to use the information provided by initial CT-scans to mark the location of the antenna in a DICOM file using a treatment planning system.

In theory, it is possible to extract the radiological regions of interest (GTV, CTV & PTV) from DICOM files when those, as defined in the DICOM RT extensions, are present, as well as custom application data that are created and read by specific manufacturers systems.

In practice however, there are many ways in which a DICOM file reader software component might fail to return correct information. Such undesired events can occur because of defects in the file reader component software or because the information contained in the file is missing or malformed. External systemic or human factors can also be involved, although those external events are beyond the awareness of the software component regarded and create issues that cannot be identified at this subsystem level.

In particular for DICOM RT files, the validity of the medical information considered is of vital importance. Returning an erroneous value could lead to the wrong patient being diagnosed, or the incorrect configuration of the size, shape and location of the tumor, hence treating a patient with unreasonable doses or in the wrong place.

Therefore, guaranteeing that the software delivers the information correctly is a safety-critical requirement for the software component.

Those considerations lead to the following question:

How to produce an effective and safe DICOM file reader software component?

Delimitations & Restrictions

Initially, the system was also intended to perform the construction a 3D model from the series of sequential 2D images stored in DICOM files acquired from the initial CT-scans. However, it rapidly appeared that the topic of multi-planar reconstruction alone was vast enough to be the subject of a single thesis.

The software considered in this study focuses on specific parts of the DICOM standard, namely the DICOM RT Information Objects Definitions (IOD) (10), data encoding (25), data dictionary (26) and media storage file format (27). Those parts involved commands, service classes, messages exchange and networking aspects of DICOM will be ignored for the development of this project and so will the various other medical modalities that DICOM applies to.

It must also be noted that the former versions of the standard (ACR-NEMA 1.0 and 2.0) did not comport a file format definition. Although it was possible to store a DICOM data stream in a file, the current file meta-information header was introduced with DICOM 3.0, as specified in part 10 (27). Although it is possible to determine the format of a file originating from those older versions of the standard, it is inelegant and relies on heuristics to determine the transfer syntax that specifies the encoding used (11).

Also, many data elements are labeled "retired" or superseded by newer elements in the current DICOM specification and they are not expected to be supported by the system. Enabling retro-compatibility with previous versions is costly to implement and will be ignored entirely within this project, unless efficient solutions are discovered with the use of Components-off-the-shelf (COTS).

Other restrictions apply to the development environment of the system, which is a Windows XP based workstation with the Visual Studio .NET framework installed for development. Toolkits intended for alternative platforms or development suites will be disregarded.

Document Outline

This chapter introduced the context and problem area and formulated the research question. A presentation of prostate cancer, radiotherapy treatment and the system under consideration was given, as well as an introduction to the DICOM standard.

In the next chapter, the literature background is reviewed. Safety-critical systems and medical devices are defined and a closer look at the DICOM standard and its challenges is undertaken.

The following chapter defines and justifies the methodology guiding the research process. The research approach and research planning are exposed.

Then the case study is outlined including the activities of data collection of tools and file samples and development and testing activities for the project. The expected outcomes for the project are established and a means for measuring and assessing the actual results is provided. The execution of the case study is then documented and its results are presented objectively as are the challenges faced during its development. Consequently, an analysis and discussion of the results conclude the study. Insights and reflections on the research knowledge gained are debated. Finally, directions for future research are pointed out.

THEORETICAL CONTEXT

This section presents a literature background review on the topic of safety as applied to medical devices. Let's point out that medical devices encompass hardware systems, as well as information systems or stand-alone software systems (28)(29).Next, a closer look at the DICOM specification is undertaken and the problems that the standard resolved along with the issues that it introduced are exposed.

Safety-Critical Medical Devices

Medical devices are complex systems composed of hardware and software whose dependability is regarded as the most important property and whose failures to perform dependably may have dramatic implications (30). Their trustworthiness ultimately defines whether they will be safely used or if they will be rejected by their users (31). The integration of software components in such systems is subject to extensive risk management, verification and validation due to the high quality requirements expected from them by the medical practitioners, patients and regulatory organizations (29)(32)(33)(34).

Safety-critical systems are defined as systems whose failure puts human health, life or environment at risk. Sommerville (31) distinguishes between primary and secondary safety-critical systems. The former has direct implications on safety whilst failures in the latter may cause harm via indirect ramifications thus rendering them harder to detect and prevent due to the unpredictable nature of the resulting side-effects.

For example, a control program for an injection pump for insulin within the patient body can cause direct harm in case of malfunction. On the other hand, a software failure, e.g. a buffer overflow may corrupt some adjacent data. That would cause no harm in itself but can eventually lead to a hazardous situation, for instance the wrong value being displayed on a monitoring device and improper actions – or lack thereof – being performed by clinicians (28)(31).

Safety is only one dimension of dependability, along with other software quality properties, namely reliability, security and availability, which are separated but related (31)(35). Those emergent properties, as well as verification and validation activities, must be considered early in the development and not as an afterthought after implementation and deployment (32)(35). The cost for modifying the design of software in service can be an order of magnitude greater than that of changes performed at the requirements specification stage(35)(36).

There has been a great deal of confusion between reliability and safety. However, whereas reliability is concerned with effectiveness and performance and the absence of all software failures, safety concentrates only on those failures that can cause harm (28).

Similarly, safety and security are two software qualities that share similar characteristics and the goal to prevent undesirable events (30). Standards, guidelines and regulatory directives for safety recommend general process-oriented approaches to use during software development and methods such as hazard analysis in order to identify and analyze all possible hazards (29)(30)(33). Bowen states that: "safety must be designed in a system and dangers must be designed out" (35). Security on the other hand relies on products-oriented techniques, where the software should exhibit certain features to combat domain-specific known threats (30)(31).

The development of critical-systems tends to be a conservative, methodical and rigid process that is reluctant to changes toward newer, less-experienced methods that have the potential to improve current systems (33).

In order to cope with this imperfect situation, federal organizations rely on risk management and control of the software life cycle process to regulate and validate systems.

Risk Management

Although regulatory organizations such as the US Food and Drug Administration (FDA) require systems to be *effective* (28)[p. 20] and *as safe as possible*, this is hard to quantify in practice. Rather than immutably defined criteria for evaluation, they rely on a set of recommendations as guidelines for development processes and current best practices of software engineering. Since guaranteeing *complete* safety is impossible, we can strive for *optimal* safety using a combination of the three concepts of risk management, quality management and software engineering (37)[p. 30].

In order to approve a system, it must be demonstrable that sufficient software validation has been carried out in a systematic, rigorous fashion. According to the FDA, validation is achievable by integrating software life cycle management with risk management activities, as recommended as Quality System regulations for medical devices systems and software in (29)[p. 1].

This calls for the implementation of the standard and technical report published by the Association for the Advancement of Medical Instrumentation (AAMI), which publishes a set of documents pertaining to software in medical device. Those documents, especially the Technical Information Report on medical device software risk management (AAMI TIR32:2004) (28) and the standard for medical device software life cycle process (ANSI/AAMI/IEC 62304:2006) (37), both partly written by members of the FDA, constitute reliable references for the development of safe systems for medical purposes.

The AAMI TIR32:2004 describes how software risk management constitutes only one stage within the global device risk management. Risk management is constituted of risk analysis, evaluation and control, as subsequently described. The accurate definition for those terms is provided in a glossary provided in appendix A. The recommendations provided in those documents are followed to conduct the software development process and achieve safety for the medical device considered in this study. The result of the risk analysis elaborated and applied to the project developed in this study can be found in Appendix B.

Risk analysis is a process during which the intended use of a device, the associated hazards for each use and their causes, and the risk of each hazard are identified. Hazards can be caused by software, hardware, user and environment events, or more often, a combination of those (31). The outcome for this stage is usually presented in the form of a table that lists harm probability and severity for each hazard and combining those to quantify their overall risk.

Risk evaluation determines whether each risk calculated in the previous step is acceptable or require control measures. For every risk greater than a certain threshold, risk control measures are planned for input to the next stage.

Risk control intends to assign risk control measures for every unacceptable risk found. Those can be preventive, corrective or mitigative in order to, respectively, avoid causes for hazards, detect when causes for hazards occur and perform corrective measures or prevent the severity of harm, should the causes happen. Another categorization divides risk control measures between inherently safe by design, protective measures and information for safety (labeling and documentation).

Finally, in order to achieve effective software safety, a holistic approach of the system and its safety requirements is necessary (28)(29)[p. 7](35)(36). Indeed, medical devices are composed of hardware and software, plus the interactions with the users and the environment must be taken into account. Even for those medical devices composed solely of software, it is needed to consider the system as a whole by capturing its intended uses, user environment, platforms and configuration management using domain and current context knowledge (28). Doing otherwise could induce the loss of safety requirements or the underestimation of risks.

The DICOM Standard: Solutions & Issues

No matter the area, standards are always a step forward for multiple reasons, including their ability to specify minimal requirements for a category of products, simplifying agreement between entities by claiming conformance to and improving the state-of-the-art by pulling upwards the minimal level of acceptance required by users (30)(35)(38). They can also relieve manufacturers from responsibility and subsequent legal suits in case of accidents when these can prove conformance to a regulatory standard (35)(38).

Prior to DICOM, medical treatment and diagnostics already improved much thanks to the introduction of IT solutions, but the procedures did not yet flow as seamlessly from beginning to end (31). Instead, there was a problem of discontinuity in medical treatments due to the lack of interoperability between systems. Operating information across multiple systems within institutions was a non-trivial task due to the usage of different formats of data and the lack of centralized databases (32)(37)[p. 135]. Those are the issues that the participating members of the DICOM committee and the manufacturers involved strive to solve(7).

Although the motivation behind DICOM is to promote interoperability, it is not sufficient for a device application to proclaim DICOM compliancy to enable plug-and-play integration. (8) emphasizes the fact that *DICOM is an enabling technology which does not standardize applications*.

As (8) and (39) point out, DICOM is voluntary standard with no conformance organization in charge of validating compliance of systems to the standard or compatibility between devices. It is the vendors' and users' responsibility to determine whether or not their systems will be compatible.

Besides conforming to the DICOM specification, manufacturers must publish a conformance statement. That document, whose template is defined by the standard, acts as a requirements specification that enables users to determine whether a piece of equipment from a manufacturer will integrate in their existing environment. However devices must be verified against their conformance statement by the manufacturers. Additionally, acquirers are suggested to validate the interoperability of the imaging equipments before they purchase those (8).

In real-world applications, however, one hundred percent conformance to the standard is the exception and not the rule. Numerous problems of conformance in the implementation of devices have been reported [42] and confirmed by professional software developers [Appendix C]. This is often caused by the complexity and thoroughness of the specification and the presence of many optional modules within DICOM objects. Another problem originates from the fact that DICOM and its application area is limited to a much applied field and targets a rather small audience. Therefore, there are few handbooks available on the topic and manufacturers must rely on the specification itself to build their systems. Overall, *DICOM 3.0 conformance is rather poor* (39).

METHODOLOGY

Research Approach

A multiple-case study will be elaborated to answer the research question. Indeed, different angles will be investigated to determine whether it is possible to produce an effective reliable DICOM file reader, and if it is, how to achieve such a software component. Since the outcome and the methods to answer the question are not a priori clearly defined, the research constitutes a study of the exploratory type (40). The data collection follows a qualitative process although it contains quantitative elements for the comparison of different software components. The cases are fully described in a separate chapter.

A case study approach is motivated by the fact that the research statement is a "How?" type of question, that it addresses modern issues (41). Qualitative case studies typically constitute empirical researches that result in having more variables than data points. This process is especially well-fit for this investigation, because the software pieces that will be evaluated possess complex characteristics and emerging properties that cannot be controlled. This would be hard to analyze with the same depth with other approaches such as surveys or experiments. It is also clearly trying to tackle some current issue in medical software; therefore case studies are preferred over histories or archival analysis.

The outcome of the research is applied, that is, it will solve some concrete problem in industry. At the same time, it is expected to provide enough abstraction to be theoretically generalizable to other projects in safety-critical systems development. Therefore it is intended to contribute to the existing body of knowledge.

Research Process

This study being of the exploratory type, relevant information sources and DICOM resources need to be identified and localized. This is achieved by following the pointers provided by the company and in the existing literature and "snowballing" from reference to reference to find useful material. More precisely, the initial sources of information consist of:

- The company documentation, fascicles, paper presentations and other informal material

- Introduction to the project by brief board presentations and inputs from the company staff and discussions with the company members using unstructured interviews or prepared written lists of questions

- Studying the existing system requirements, specification and architecture and other research papers of the company

- Articles from journals, books and web pages

- Interview with a software developer familiar with DICOM working a hospital to gain precious insights from the practical experience of professionals in the field. This will be conducted as a semi-structured interview consisting of a prepared set of closed questions as well as more open ones.

DICOM remaining a specialized category of medical software, there is a limited number of reliable sources provided by experts in the field referenced by many other authors. Moreover, as there is little paper documentation regarding DICOM, most of the research is carried out on the Internet and from contacts at the company and the hospital.

Subsequently, the case study documented further in this report is soundly designed using the knowledge acquired. It contains a comparison of software components according to certain evaluation criteria for effectiveness and reliability that are defined on the basis of an initial risk analysis. The data collection is performed at that stage too, which includes building a prototype and gathering and evaluating existing systems.

The project environment will be within the company itself, *Micropos*, that wants to integrate a new functionality into their existing system. Upon the signature of a confidentiality agreement, unrestricted access to the company resources, documentations and support is provided. My first task as a researcher will be to gain some basic knowledge about the organizational structure of the company, their current processes, their existing *4DRT* system and the future directions for development. The company will gain insights in development methods and improvement from the research and I will make use of the company resources and current settings as foundation for my research.

The company is fairly small in terms of headcount and promotes an open door policy and face-to-face communication. That plus the uncertainty involved with such exploratory research and the likelihood of rapidly changing requirements as the new information becomes available and the project evolves are all good arguments in favor of rapid development methods. Agile software development and in particular *Scrum* will be used to drive the software project development aspects of my research at the pace of short continuously improving iterations.

Scrum (42) is an iterative incremental process for managing software development that defines practices and roles. It has proved successful even for larger projects; preferring quick and direct communication and rapid delivery of small additions over monolithic plan-to-release developments. Indeed, the responsibility is slightly shifted from the managers to the project team. Shortly put, it is a management process that is fairly easy to implement.

This development method is well-fit for this project, because it will provide flexibility and adaptability to support my exploratory research within a socio-technical environment that involves many stakeholders from various specialties. It will help me gather initial information via the company presentation and documentation, and build knowledge around it in an incremental fashion as the project unfolds. It brings valuable asset for me, as I have little to no previous experience in the development process of medical device software.

Although it is part of agile methods and as such is intended to relieve the project management from rigid heavy processes, Scrum does not represent a threat to the formalism required in the development of medical software and will be kept at a high-level to direct the project development process.

CASE STUDY

The practical part of this study is constructed to experiment on how to find or design and implement an effective DICOM files reader and how to test and validate it sufficiently. Such a software component must be capable of reading and extracting patient demographic information as well as radiotherapy relevant data (location of ROIs, organs at risk and other fields specific to *Micropos*' system) from a DICOM file without failing in a way that can lead to unsafe situations. The resulting software component needs to be integrated within the whole existing *Micropos*' four-dimensional real-time radiotherapy system. A multiple-case study is elaborated to obtain the necessary empirical knowledge and therefore answer the research question.

The Cases

For the first case, I will create a homemade prototype to gain hands-on experience about DICOM implementations and to experiment how strictly DICOM files respect the standard. A very simple DICOM file reader will be developed by following the specification of the file structure defined in the standard as strictly as possible. Although basic, it is expected to be capable to parse any DICOM file and display the values of its DICOM Data Elements in a text readable form. The relevant parts of DICOM to build such a system are the ones dealing with data encoding (PS 3.5), data dictionary (PS 3.6) and file format (PS 3.10).

For the second case, I will search for existing components-off-the-shelf (COTS) to use them as-is or adapt them. My aim is to collect DICOM-related software programs, libraries, source codes and architectural frameworks that are publicly available, some of which have already been discovered during the literature review (43)(44)(45). I will meticulously gather as many of them as possible and add them in a list in order to keep track of their origin and characteristics.

The COTS will be filtered through a number of selection criteria described below to determine whether they fulfill the functional requirements of the system. This will isolate the ones that match the intended use from the ones that are not fit for purpose. Practically, I will browse and check the COTS from the list sequentially and mark the components that meet the criteria for later inspection and evaluation.

It is likely that I will thereafter need to interface, encapsulate or adapt suitable existing COTS before they can be incorporated into *Micropos*' system. Plans to develop a fully functional custom component from scratch if none of the existing software COTS are appropriate are also considered.

It must be noted that by being provided as-is for free or trial versions, publicly available COTS are not liable to exhibit optimal reliability. Therefore relevant concerns are the time and costs involved in evaluating and adapting the software, as well as the effort of testing involved, considering that medical devices have high safety requirements. This is where the activities of risk management described in the previous chapter take form: at this point I will proceed to evaluate the selected one(s) in terms of the quality requirements based on an initial risk analysis.

Verification of the resulting solution, provided that such a component is achievable, will be carried out by testing the executable program against a variety of DICOM files. In order to accomplish this, a sufficient amount of DICOM files from different modalities will be gathered from both online and clinical sources and used as input for system-level testing. Those file samples must represent the wide diversity of all existing DICOM files from various modalities, image formats, dimensions and sizes, including ones presenting unusual or abnormal data structures whenever found. Besides public samples from internet sources, a number of DICOM files will be requested from the hospital, although those real-world files need to be anonymized for privacy reasons.

Once a software component satisfying the quality requirements is achieved, it will have to be integrated in the existing *Micropos*' *4DRT* system and interface with it. Note that the component does not present a GUI; therefore no direct interaction with clinical personal is required. Such a system that cannot lead to a direct hazard is ranked as *secondary critical* by (30) and (38).

Selection Criteria

Before the activities of implementation or adaptation, a preliminary step is to thoughtfully define the parameters according to which the DICOM software COTS will be rated in terms of their effectiveness and suitability. The selection is based on the availability, functionality, platform of operation, modes of utilization and adaptability of the DICOM software COTS. A set of criteria guaranteeing effectiveness is constructed based on their technical characteristics and the general features that they exhibit.

First of all, there must be a free or trial version of the component accessible on the Internet so that it can be retrieved. The URL leading to the program must be valid and the website up and running, which is obvious but not always the case. There is also the above-mentioned logistical constraint about the operating environment: the programs that don't run on a Windows XP computer or not implemented in .NET compatible languages are ruled out.

Moreover, the program must be adaptable, which means either that its source code must be available or that it can be integrated in the system under the form of a library object (ActiveX, DLL) or at least executable with command line (.exe) if it exhibits some public API to manipulate it. Higher level architectural frameworks without an actual executable implementation or source code represent a valid alternative as well.

Finally, the component must fit its intended purpose. The main feature that is required is the ability to read information from DICOM files and validate them doing so. Given a file and the name or number of a specific attribute as input, the component should be able to determine whether the file is a DICOM file, whether it is correctly structured and then parse it to read and return the value associated with the attribute. The tool also needs to offer the possibility to be readily extendable for later increase of functionalities.

Evaluation Criteria

The evaluation criteria focus on those qualities related to reliability and are used to review their correctness, validity and safety. To accomplish this, risk analysis management based on the intended use of the system is performed. This process starts with a risk analysis to identify and assess the potential hazards that may occur and that the component must be able to handle (see Appendix B). The format of that risk analysis is based both on the model provided in an appendix of [13] and on *Micropos*' operatory documentation.

Expected Outcome

By experimenting with DICOM and the associated software, I will gain practical insights in the process of developing medical software and knowledge about the issues and challenges still facing those fields.

The expected outcomes from this research are:

- Knowledge about DICOM, safety-critical systems and medical devices

- A snapshot of the current DICOM context for software development

- A prototype of DICOM Reader program

- A comprehensive listing of existing DICOM software COTS

- The selection of one or more of those COTS to be adapted to build an effective DICOM reader and the evaluation & verification of the most suitable of those COTS to produce it

- The implementation of an effective DICOM Reader software component, provided that it is achievable in the limits of this study. Such a component shall conform to all the criteria established and handle all the hazards identified during risk analysis.

RESULTS

In this section are presented the results of study of the cases that have been carried out. They tend to confirm, along with the interview transcribed in Appendix C, the hypothesis that a lack of conformance to the DICOM standard exists. However, the production of the DICOM File Reader component needed by Micropos could be achieved with effectiveness and reliability.

DICOM Reader Prototype

I began with the building of a prototype of a simple DICOM parser (pictured on Figure 4 on page 8) according to the rules specified in parts PS 3.5 (data encoding) and PS 3.10 (file format) of the standard. I left out a few details such as reading some data types simply as bytes without attempting to interpret their content.

Although basic, my program of a few hundred lines of code was designed to handle and read any DICOM file conforming to the standard, as in Figure 4, and raise exceptions whenever data elements would not respect their formal specification.

Especially, I was meticulous when it came to checking the conformance with the specification or lack thereof, as investigated in (39), and I performed strict verification on the length and formats of the data elements encoded in the files (See PS 3.5).

For instance, in part PS 3.5 of the standard, a data element with a Value Representation of "DA" is a Date type that is formally defined in DICOM as

> "A string of characters of the format *yyyymmdd*; where *yyyy* shall contain year, *mm* shall contain the month, and *dd* shall contain the day. This conforms to the ANSI HISPP MSDS Date common data type. For reasons of backward compatibility with versions of this standard prior to V3.0, it is recommended that implementations also support a string of characters of the format *yyyy.mm.dd* for this VR. The characters allowed are '0' to '9' and the dot '.'. The length of the data element is fixed and shall be either 8 or 10."

Surprisingly, for an encoding rule relatively straightforward like this one, some of the first DICOM files provided by *Micropos* that I used as input to execute my program did not follow the specification. For example, they used a format such as *yyyy/mm/dd* instead of the required one presented above.

DICOM Software

The outcome of the second case study is a list of DICOM software components that is expected to be quite comprehensive, if not exhaustive, and the finding of a powerful and effective tool to implement DICOM systems, manipulate DICOM files and therefore address *Micropos*' needs.

The list is constituted of all the 190 DICOM-related software tools that I have encountered, regardless of their origins and functionalities. From there, I began establishing a distinction between them in consideration with my selection criteria and filled the list with their individual characteristics, resulting in the table provided in Appendix D.

Although an extensive number of DICOM tools were enumerated, it has not been my intention to inspect all of them thoroughly, because of their quantity. Rather, I chose to direct my time and effort towards the ones that seemed of interest by subjective appreciation, i.e. those originating from more recent websites, provided with complete documentation or supported by a community of developers. Those were the ones that I proceeded to acquire for further inspection, discarding the others until running out of options.
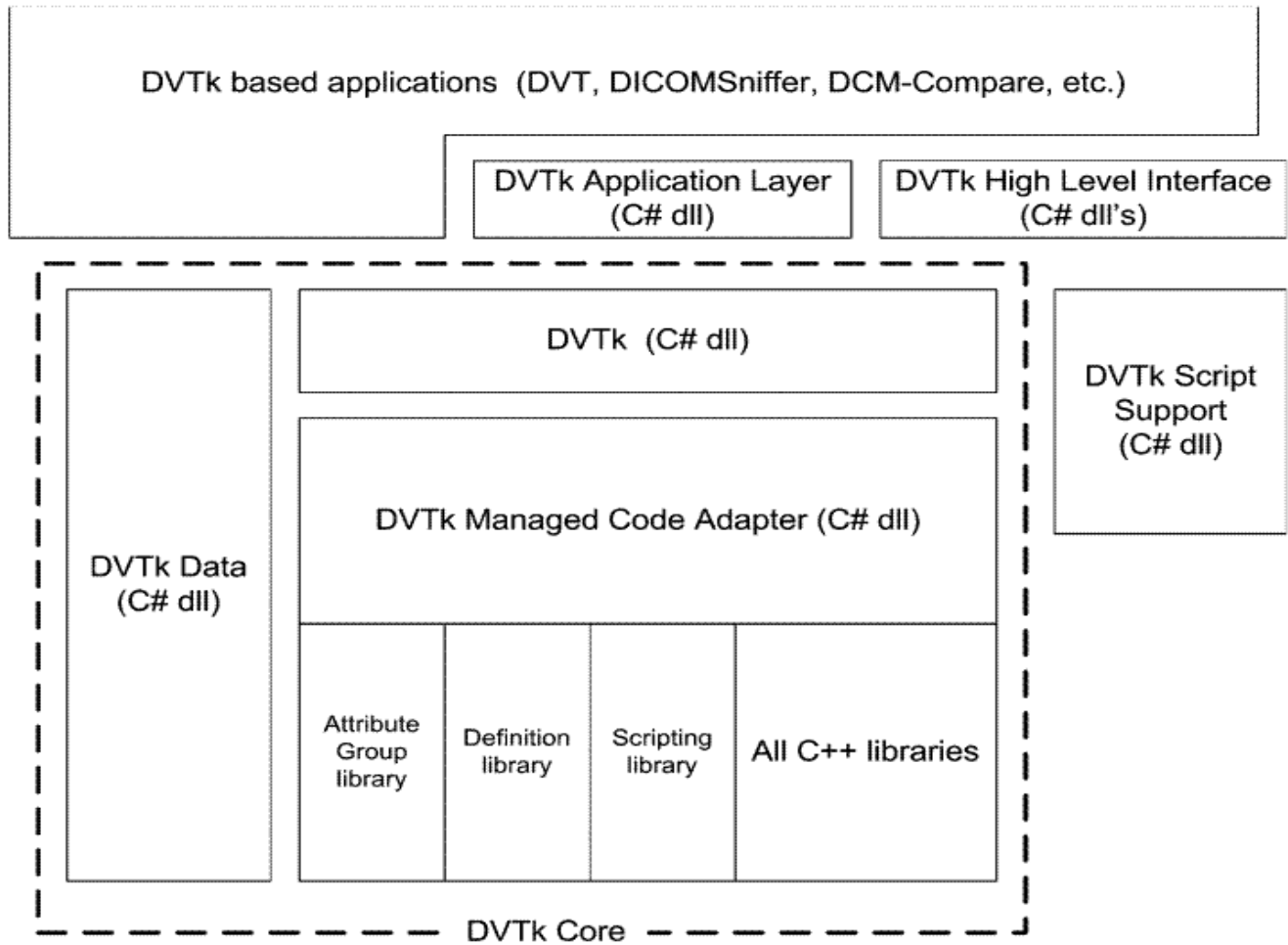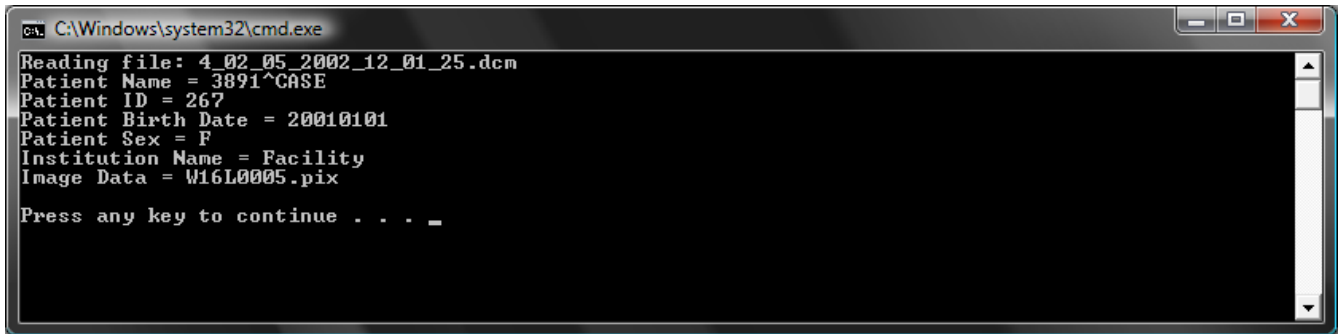
Figure 5: DICOM Validation Toolkit Component Model (45)

As a result, among the numerous COTS gathered, there were few suitable to be integrated into *Micropos' 4DRT* system. I was searching for programs available as importable libraries or with command lines interface, source codes or architectural framework but most of the tools encountered (listed in Appendix D) came with a graphical interface but no application program interface or reusable code.

Among the few suitable tools, one caught my attention as of the moment I came across it in an article written by its creators (45). Product of a recent (2006) collaborative endeavor by AGFA and Philips, the DICOM Validation Toolkit (DVTk) is an entire open source framework based on a layered architecture (Figure 5). This modularity enables anybody to freely extend it by building high-level DICOM applications on top of a common core. One can also tailor that core or adapt existing DVTk-based high-level applications that come along with the toolkit to one's own needs.

Moreover, it provides a mechanism for the validation of the files, as suggested by its denomination. That validation mechanism is implemented as a set of definition files that are a translated version of the formal specification of the Information Object Definitions established in part 3 of the standard. Those files are used by DVTk-based applications to guarantee that the structure of a Data Set contained in a DICOM file corresponds to that defined by the standard. All the necessary Information Entities, Modules and Data Elements constituting the Data Set need to be present in the file and correctly formed in order for it to be read. However DVTk is also capable of guessing the Transfer Syntax and read files exhibiting unusual data structure.

Figure 6: Micropos DICOM Reader used by a console application on a single file

In addition, the fact that it was initiated by notorious manufacturers and that it is open source means that there is an experienced and active community to support it. In fact, anyone can contribute to improve its quality or add new features and share high-level applications.

Those characteristics seem to make DVTk a suitable fit as a tool on top of which the DICOM Reader desired by *Micropos* can be built. Based on the results of the tests of the resulting DVTk-based component detailed below, the software-related risks established in Appendix B were appropriately addressed, which confirms that observation.

DICOM Files

Over 12,000 DICOM files (8 Gigabytes of data) were downloaded from 20 different web sources to test the DICOM software, of which the complete list can be found in Appendix E. The distribution of the files samples is quite diverse in terms of size, types of images (2D, 3D, 4D and even 5D), modalities by which they were produced (RT, MR, etc.) and peculiarities to guarantee adequate coverage.

Custom DICOM Reader Component

In order to fulfill *Micropos'* need to develop and integrate a DICOM file reader into their *4DRT* system, I strived to adapt the DVTk library to create a custom DICOM component that could be inserted into the main system. Accordingly, I designed an extensible program based on the DVTk core able to parse DICOM files and read the Data Elements within. The result is a simple set of .DLL files that can be referenced from any .NET based application.

The fact that it relies on DVTk means that it can perform validation for the files and thus verifies that the files are correctly structured – and that it is indeed a DICOM file before any access to the Data Elements is attempted. It also signifies that it reuses its methods for loading and reading files, which are flexible enough to enable the user to access any Data Elements within the file based on its Tag Number, even for nested Data Elements as the tests confirmed.

Evaluation, Test & Verification

The reliability of the Micropos DICOM Reader was assessed through a series of tests, including verification that the hazards identified during risk analysis are correctly avoided by the software.

First to perform system-level testing I created a separate console application that imports the Micropos library and reads a DICOM file then outputs the values of a few Data Elements chosen arbitrarily, when those are present. This is demonstrated in Figure 6 where the Patient Name (here anonymized), Patient ID, Patient Birth Date, Patient Sex and Institution Name read from one DICOM file are displayed. The other Data Elements that are not present in that particular file are rightfully not output.

Figure 7: Testing of the Micropos DICOM Reader using a batch of DICOM files as input

Another test involved building a similar console application to browse by recursion through the directory structure of my collection of DICOM files to read all the files inside. Again, the program was just meant to display a few selected Data Elements for every file with the objective to find out whether the library correctly distinguished between DICOM and non-DICOM files regardless of their file extensions. The expected behavior is that it should read the former while skipping the latter, and also that it does not crash the program despite the large batch of files input (Figure 7).

Finally, I used some specific files to accomplish more in depth verification: files using compression, missing the File Meta-Information Header and presenting private Data Elements. The result of those verifications were more than satisfactory, since all the risks listed in the table in Annex B catchable at the software level were correctly managed by the component. Apart from a few deprecated and poorly structured exceptions, all the DICOM files could be parsed, in particular unusually ones using data compression and those without File Meta-Header Information to indicate the Transfer Syntax used to decode the information in the file.

Because at the time of this writing, *Micropos'* system *4DRT* now renamed to *RayPilot* was still in development, the integration of the Micropos DICOM Reader library in the whole system was not performed. However partial integration testing was achieved thanks to the console applications interacting with the component.

CONCLUSION

One of the early outcomes of my study, supported by literature review and interview, is that there is still much work needed to make DICOM manufacturers' conformance a reality and not only a possibility. One hundred compliant DICOM files are still the exception and not the rule. I can vouch for this from personal experience and I believe that anybody who has worked long enough with the standard can back up this statement, as demonstrated by the interview in Appendix C.

Indeed, during the development of my initial DICOM file reader prototype, it appeared that all the files do not comply with the standard as strictly as intended. That observation gave me useful practical insights about the actual issues at hand and the lack of conformance of the DICOM files mentioned in the literature. This problem comes from the medical device manufacturers building DICOM systems that often do not respect the specification as closely as they are expected to.

Another issue is that many files were produced using older versions of the standard, which did not specify a file format at all. Indeed, the file structure relies on a certain Transfer Syntax defined in a section of the DICOM file called file meta-information header, but in practice, these can be missing altogether.

This results in a variety of DICOM files composed of various non-standard attributes and unusual transfer syntaxes. Such a wide range of different implementations introduces an additional complexity whose ramifications are difficult to grasp. A professional software developer in the field admitted that it is impossible in practice to develop a DICOM reader capable of reading all existing diversities of DICOM files (see Appendix C).

However, DVTk offers to address the issues previously mentioned: as approved by the tests, it is a reliable and flexible tool for developers interested in DICOM. Although as recent as 2006 and not mentioned in older articles on DICOM, it seems very promising. That is why reusing DVTk as a complete and advanced framework for DICOM software has greatly facilitated the development of my custom DICOM Reader library. Plus, the definition files being physically independent files, they are in theory the only part of the application to update to reflect the current specification as the standard evolves. This brings valuable leverage in terms of time and development costs.

As previously mentioned, *DICOM is a voluntary standard and as such, no organization for conformance approval or validation exists* (8) (39). It is still a young standard and a field that needs work and improvements in the manner it is implemented by the medical devices manufacturers. Hopefully, recent initiatives like the DICOM Validation toolkit will help pull the standard upwards and enable the developers to achieve systems of better quality.

Thus, it is indeed possible to develop and integrate a DICOM file reader software component within a 4D radiotherapy positioning system, although one must be aware of the difficulties involved with the development of such a component and the challenges still facing DICOM and its implementations by the medical device manufacturers across the world.

LIMITATIONS AND FUTURE WORK

This chapter highlights some of the difficulties and limitations encountered during the research that could imbalance the validity of the results; then, directions for future work are suggested.

Firstly, I must admit that process of conducting exploratory research was rather difficult to carry out; because I had to limit the amount of time and efforts I could spend in diverse research directions. For instance, I gathered a large quantity of DICOM software in a list (see Appendix D), but I could not afford to try them all out. Instead, I focused my attention on the few ones that seemed to exhibit the qualities that I searched and swiftly dismissed the ones that did not.

Another constraint pertains to the platform of operation, which is a Windows-based computer running the Microsoft .NET framework. Quantities of the software COTS listed in Appendix D runs on different systems and environments (Unix-based OSs, MacOS X) or written in non-supported language (e.g. Java). Although some of them appeared interesting in terms of the qualities required, they were discarded from further investigations.

Also, as previously mentioned, the DICOM standard specification is long and tedious to come to grip with in its entirety. Although I have spent a significant amount of time familiarizing myself to the whole standard, I have solely exploited a few parts of it in my project. Those parts covered by the tool that I have built represent merely a minority of its spectrum. For the purpose of my research, I needed to read and extract general patient information and few specific Radiotherapy-related data from RT DICOM files, so I didn't linger on complete DICOM solution development.

Hence, the Micropos DICOM Reader does not handle network protocols and messages exchange that are supported by DICOM as a means to retrieve RT information from PACS at a hospital. Neither is it initially able to display the images contained within the files, let alone generate videos or reconstruct 3D models of multi-frames types of pictures. These features will likely be of concern when extending the component in the future.

As mentioned in the results chapter, the actual integration of the DICOM reader is left for *Micropos* to carry out as it is unclear which attributes related to radiotherapy are to be retrieved from DICOM or whether custom private Data Elements will have to be created at the clinical level. Hopefully, the library and its documentation for operation and extension with which I provided *Micropos*, allows the company to tailor the component effortlessly to their future needs.

Finally, some functionalities of DVTk on which the *Micropos* DICOM Reader is built can be called to output information when reading a file. This can be used to display feedback related to the exceptions and errors occurring or to implement a return code mechanism to interact with *4DRT*.

BIBLIOGRAPHY

1. **Socialstyrelsen, The National Board of Health and Welfare.** *Cancer Incidence in Sweden 2006.* Official Statistics of Sweden. 2007. 978-91-85483-81-5.

2. **Dobbs, J., Landberg, T.** Clinical overview of geometric uncertainties in radiotherapy. *Geometric Uncertainties in Radiotherapy.* s.l. : British Institute of Radiology, 2003, pp. 5-10.

3. *Geometric Uncertainties in Radiotherapy.* **Wilkinson, J. M.** Manchester, UK : s.n., 2004, The British Journal of Radiology, Vol. 77, pp. 86-87. 10.1259/bjr/25924254.

4. **Schoster, C., Syrén, H.** *Integration of a realtime implant positioning system for 4D radiotherapy cancer treatment.* Signals and system, Chalmers University of Technology. Göteborg, Sweden : Chalmers University of Technology, 2005. pp. 1-62. EX078/2005.

5. *Digital technologies and quality improvement in cancer surgery.* **Mutter, D., Bouras G., Marescaux, J.** 6, 2005, European journal of surgical oncology, Vol. 31, pp. 689-695. 0748-7983.

6. **Micropos.** Micropos - MICROPOS MEDICAL. *Micropos.se.* [Online] http://www.micropos.se/.

7. **ACR-NEMA Committee.** *Digital Imaging and Communications in Medicine (DICOM) Part 1: Introduction and Overview.* National Electrical Manufacturers Association. 2008. Standard specification.

8. *DICOM : Your guarantee to interoperability?* **Oosterwijk, H.** [ed.] Bellingham, WA, USA Society of Photo-Optical Instrumentation Engineers. Newport Beach, CA, USA : SPIE, Bellingham, WA, USA, 1997. PACS design and evaluation : engineering and clinical issues. Vol. 3035, pp. 165-169. 0-8194-2446-3.

9. **NEMA.** DICOM Homepage. *NEMA.org.* [Online] dicom.nema.org.

10. **ACR-NEMA Committee.** *Digital Imaging and Communications in Medicine (DICOM) Part 3: Informatoin Object Definitions.* 2008. Standard specification.

11. **Clunie, D. A.** *Medical Image Format FAQ.* [Online] December 7, 2008. http://www.dclunie.com/medical-image-faq/html/index.html.

12. *Introduction to the DICOM standard.* **Mildenberger, P., Eichelberg, M., Martin, E.** 4, April 2002, European Radiology, Vol. 12, pp. 920-927. 1432-1084.

13. *The DICOM standard. A breakthrough for digital information exchange in cardiology.* **Parisot, C.** 3, September 1995, The International Journal of Cardiac Imaging, Vol. 11, pp. 171-177. 0167-9899.

14. *Introduction to the ACR-NEMA DICOM Standard.* **Bidgood, W.D. Jr, Horii, S. C.** 1992, RadioGraphics, Vol. 12, pp. 345-255.

15. *PACS mini refresher course. Network and ACR-NEMA protocols.* **Horii, S. C., and Bigwood, W. D. Jr.** 1992, RadioGraphics, Vol. 12, pp. 537-548.

16. *A nontechnical introduction to DICOM.* **Horii, S. C.** 1997, RadioGraphics, pp. 1297-1309.

17. **Horii, S.C., Prior, F.W., Dean Bidgood, Jr., W., Parisot, C., Claeys, G.** DICOM: An Introduction to the Standard. [Online] 1993. http://www.dicomanalyser.co.uk/html/introduction.htm.

18. *DICOM demystified: A review of digital file formats and their use in radiological practice.* **Graham, R. N. J., Perriss, R. W., and Scarsbrook, A. F.** 11, 2005, Clinical Radiology, Vol. 60, pp. 1133-1140. 10.1016/j.crad.2005.u.003.

19. *Understanding and Using DICOM, the Data Interchange Standard for Biomedical Imaging.* **Bidgood, W. D. Jr, Horii, S. C., Prior, F. W., Van Syckle, D. E.** 3, June 1997, Journal of the American Medical Informatics Association, Vol. 4, pp. 199-212.

20. **Derhy, Serge.** Dicom Tutorial. *Eviewbox DICOM java applet.* [Online] [Cited: May 1, 2008.] Paris, France.

21. *Clinical importance of the DICOM structured reporting standard.* **Bidgood Jr, W.D.** 5, October 1998, The International Journal of Cardiac Imaging, Vol. 14, pp. 307-315. 10.1023/A:1006073709957.

22. *DICOM Structured Reporting part 1. Overview and Characteristics.* **Hussein, R., Engelmann, U., Scroeter, A., and Meinzer, H.-P.** 2004, RadioGraphics, Vol. 24, pp. 981-986. 10.1148/rg.243035710.

23. *A model of DICOM-based electronic patient record in radiation therapy.* **Law, M. Y. Y.** 2-3, Computerized Medical Imaging and Graphics, Vol. 29, pp. 125-136. 10.1016/j.compmedimag.2004.09.012.

24. **Sethi, H. R.** DICOM Beyond 2000 - Current Status of DICOM RT Extensions. *The American Association of Physicists in Medicine.* [Online] 2001. [Cited: February 2, 2009.] http://www.aapm.org/meetings/2001AM/pdf/7176-98620.pdf.

25. **ACR-NEMA Committee.** *Digital Imaging and Communications in Medicine (DICOM) Part 5: Data Structure and Encoding.* 2008. Standard specification.

26. **ACR-NEMA Committee.** *Digital Imaging and Communications in Medicine (DICOM) Part 6: Data Dictionary.* 2008. Standard specification.

27. **ACR-NEMA Committee.** *Digital Imaging and Communications in Medicine (DICOM) Part 10: Media Storage and File Format for Media Interchange.* 2008. Standard specification.

28. **AAMI.** *Medical device software risk management.* [ed.] Association for the Advancement of Medical Instrumentation. s.l. : Association for the Advancement of Medical Instrumentation, 2004. 1-57020-233-8.

29. **U.S. Department of Health and Human Services, Food and Drug Adrministration.** *General Principles of Software Validation; Final Guidance for Industry and FDA Staff.* U.S. Department of Health and Human Services, Food and Drug Adrministration, FDA. 2002.

30. *An Analysis of Selected Software Safety Standards.* **Wallace, D.R., Kuhn, D.R., Ippolito, L.M.** [ed.] Gaithersburg, MD Nat. Inst. of Stand. & Technol. Gaithersburg, MD, USA : s.n., 1992. Proceedings of the Seventh Annual Conference on Computer Assurance (COMPASS '92). pp. 123-136. 0-7803-0579-5.

31. **Sommerville, I.** *Software Engineering 8.* s.l. : Addison Wesley, 2006. p. 840. 0-321-31379-8.

32. *High-Confidence Medical Device Software and Systems.* **Lee, I., Pappas, G.J., Cleaveland, R., Hatcliff, J., Krogh, B.H., Lee, P., Rubin, H., Sha, L.** 4, April 2006, Computer, Vol. 39, pp. 33-38. 0018-9162.

33. *Assurance and Certification of Software Artifacts for High-Confidence Medical Devices.* **Jones, M. P.** [ed.] University of Pennsylvania Computer and Information Science. Philadelphia, USA : s.n., 2005. High Confidence Medical Device Software and Systems (HCMDSS) Workshop.

34. *Enabling Certification through an Integrated Comprehension Approach.* **Jetley, R., Purushothaman Iyer, S.** [ed.] University of Pennsylvania Computer and Information Science. Philadelphia, USA : s.n., 2005. High Confidence Medical Device Software and Systems (HCMDSS) Workshop.

35. *Safety-critical systems, formal methods and standards.* **Bowen, J. P., Stavridou, V.** 4, July 1993, Software Engineering Journal, Vol. 8, pp. 189-209. 0268-6961.

36. *Formal Methods and Software Safety.* **Bowen, J. P., and Stavridou, V.** [ed.] Heinz Frey. Zurich, Switzerland : Pergamon Press, 1992. Safety of Computer Control Systems 1992 (SAFECOMP'92). pp. 93-98.

37. **AAMI.** *Medical device software - Software life cycle processes .* [ed.] Association for the Advancement of Medical Instrumentation. s.l. : Association for the Advancement of Medical Instrumentation, 2006. Vol. ANSI/AAMI IEC 62304:2006. 1-57020-258-3.

38. *Formal Methods in Safety-Critical Standards.* **Bowen, J. P.** Brighton, UK : IEEE Computer Society Press, 1993. Proc. 1993 Software Engineering Standards Symposium (SESS'93). pp. 168-177. 0-8186-4240-8.

39. *Quality of DICOM header information for image categorization.* **Güld, M. O., Kohnen, M., Keysers, D., Schubert, H., Wein, B. B., Bredno, J., Lehmann, T. M.** [ed.] H. K. Huang Eliot L. Siegel. San Diego, CA, USA : SPIE, 2002. Medical Imaging 2002: PACS and Integrated Medical Information Systems: Design and Evaluation. Vol. 4685, pp. 280-287. 10.1117/12.467017.

40. Theory of Research. *Thames Valley University.* [Online] http://brent.tvu.ac.uk/dissguide/hm1u1/hm1u1text2.htm.

41. **Yin, R. K.** *Case study research: design and methods.* Second Edition. s.l. : SAGE Publications, Inc., 1994. Vol. 5. 0-8039-5663-0.

42. Scrum (development). *Wikipedia.* [Online] 2008. [Cited: February 2, 2009.] http://en.wikipedia.org/wiki/Scrum_(development).

43. *Free DICOM Image Viewing and Processing Software for Your Desktop Computer.* **Escott, E. J., Rubinstein, D.** 5, 2003, RadioGraphics, Vol. 23, pp. 1341-1357. 0271-5333 .

44. *DICOM Shareware: A Public Implementation of the DICOM Standard.* **Moore, S. M., Beecher, D. E., Hoffman, S. A.** 1994. Medical Imaging 1994: PACS: Design and Evaluation. Vol. 2165, pp. 772-781. 10.1117/12.174371.

45. *Mastering DICOM with DVTk.* **Potter, G., Busbridge, R., Toland, M., Nagy, P.** Supplement 1, November 2007, Journal of Digital Imaging, Vol. 20, pp. 47-62. 1618-727X.

APPENDIX A: GLOSSARY & DEFINITIONS

4DRT: Four-dimensional Radiotherapy

AAMI: Association for the Advancement of Medical Instrumentation

ACR: American College of Radiology

COTS: Components-off-the-shelf

CT: Computed Tomography

CTV: Clinical Target Volume

DICOM: Digital Imaging and Communications for Medicine

DICOM Data Element: Attribute defining an DICOM Information Object

DICOM Data Set: Structured set of DICOM data elements e.g. encapsulating an Information Object Instance

DICOM Data Stream: Data set encoded according to the rules defined in the fifth part of the standard (25)

DICOM Information Object: Object-Oriented representation of a real-world entity, such as a patient, study, RT image, etc.

DICOM RT: DICOM applied to the RT Modality including specific information such as ROIs

DVTk: DICOM Validation Toolkit. Open source framework developed by AGFA and Philips

FDA: Food and Drug Administration

GTV: Gross Tumor Volume

Hazard: Potential source of harm

Harm: Damage to the health of people, their properties or environment

HIS: Hospital Information System

Modality: Class of medical image acquisition equipment, such as RT, CT, MR, Ultrasound, etc.

MPR: Multi-Planar Reconstruction

MR: Magnetic Resonance

NEMA: National Electrical Manufacturers Association

PACS: Picture Archiving and Communications System

PTV: Planning Target Volume

RCM: Risk Control Measure

RIS: Radiology Information System

Risk: Combination of probability and severity of harm

ROI: Region Of Interests

RT: Radiation Therapy aka Radiotherapy

RTPS: Radiation Treatment Planning System

TS: Transfer Syntax

VR: Value Representation

APPENDIX B: RISK ANALYSIS

External & System Related Hazards	Probability	Severity	Overall Risk	Risk Control Measure
Interface misconception, the system interprets data wrong (e.g., typecasting)	Medium	Medium	Medium	Interface design + Integration testing
Wrong DICOM file input into the system	Low	High	Medium	Double check with practicians
File input is not a DICOM file	Low	Low	Low	Software must be capable of detecting it

Software Related Hazards	Probability	Severity	Overall Risk	Risk Control Measure
Poor buffer management causes overflow	Medium	Low	Low	Memory management + Software verification
Software defects	High	Medium	Medium	Software Verification
Software crashes	Low	Medium	Low	Software Verification & Validation
Software cannot handle all types of Transfer Syntaxes	Medium	Low	Low	Shall be designed into the software using the standard specification
Software is not able to read specific nested RT data elements	Medium	Low	Low	Shall be designed into the software using the standard specification

DICOM File Related Hazards	Probability	Severity	Overall Risk	Risk Control Measure
File data corruption	Low	High	Medium	If detectable, return appropriate error code Else double check value with radiology expert
Missing RT data	Low	Low	Low	Return appropriate error code
Data use wrong encoding scheme (date)	High	Medium	Medium	Software designed to handle small variations in data encoding
Header TS not Explicit VR – LittleEndian but old standard Implicit Little	Low	Low	Low	Software designed to detect this and attempt to distinguish the correct TS
Lack of Data Dictionary	Low	Low	Low	Continue parsing if VR is Explicit Otherwise return error code
Data element not recognized	Medium	Low	Low	Non-critical values are ignored. No warming necessary
Missing File Meta Information Header	Low	Low	Low	Software designed to detect this and attempt to distinguish the correct TS
The TS for the Data Set is not the one defined the File Meta Info Header	Low	Low	Low	Software designed to detect this and attempt to distinguish the correct TS
Large file size	Medium	Low	Low	The program shan't read the image pixel data
Large header size (e.g. contains overlay information or thumbnail images)	Medium	Low	Low	Use a buffer with limited size and loop for read accesses
Transfer Syntax error => file becomes unreadable from a certain point	Low	Low	Low	Software designed to detect this and attempt to distinguish the correct TS and return appropriate error code if it is not possible
Right patient but wrong RT data	Low	High	Medium	Double check with radiology expert
Other file peculiarities goes undetected	Low	Medium	Low	Return appropriate error when any item does not strictly conform to the standard
Data uses compressed Transfer Syntax	Medium	Low	Low	Software designed to detect and handled this as specified by the standard

Risk analysis for the DICOM software component for *Micropos*. Table format based on (28) and *Micropos* documentation.

APPENDIX C: INTERVIEW WITH A DICOM SOFTWARE DEVELOPER

Dan Sandberg is a software developer involved in clinical practices and the DICOM standard for over 10 years. He agreed to answer a few questions regarding the DICOM standard and the integration of such a system in the medical process in regard to safety considerations.

The following is the report of the discussion.

Q: There is a lack of conformance in DICOM files to the standard (e.g. bad encoding, transfer syntax that switches in the middle of the file). Have you noticed that yourself?

A: Yes

Q: Is it practically possible to implement a file reader that can handle all different types of files?

A: Not really, you would have to be aware of all existing manufacturers' implementations.

Q: Has there been progress in conformance to the standard in the past 10 years?

A: The standard is constantly evolving.

Q: Do you still come in contact with old files conforming to previous versions of the standard (i.e. presenting no file meta-information header)?

A: Yes, but that's rare.

Q: The system in its first version will use files as input, like on a USB-key or such, can this process introduce errors like fetching the wrong patient information?

A: That's possible, but not likely if you rely on the patient ID or study date for identification.

Q: Future version of the system will have to connect to a PACS to retrieve images. Is there a need for a DICOM conformance statement for such a system?

A: Maybe if the system sends images.

Q: Do all HIS software systems connecting to a PACS provide a conformance statement?

A: All systems should have one.

Q: How do you fetch DICOM files? Based on patient information or content-based retrieval?

A: You have to know the patient ID or session ID maybe.

Q: Can you see any safety-related risks involved in the medical process of using a DICOM file reader?

A: Risk of wrong images maybe? That's hard to say, it's really case by case.

APPENDIX D: DICOM SOFTWARE COTS

Software Name	Status	Platform	Language	Type	Source	Format	Software URL
3D-DOCTOR	$	-	-	Image	-	-	http://www.ablesw.com/3d-doctor/
3DVIEWNIX	-	Sun/Unix	-	-	-	-	http://www.mipg.upenn.edu/~Vnews/index.html
AccuImage	Offline	-	-	-	-	-	http://www.accuimage.com/
AccuSoft ImageGear	Demo		Many				http://www.accusoft.com/products/imagegear/ignet/version_history.asp
AccuSoft ImageGear MD	Demo	Win	.NET			SDK	http://www.accusoft.com/products/imagegear/igmdnet/overview.asp
ACTIV 2000	Free	Win	-	?	No	GUI	http://193.55.58.11/Activ2000/activ2000.htm
Alice	$	-	-	-	-	-	http://www.perceptive.com/
Alitech	Offline	-	-	-	-	-	http://www.alitech.com/
AMIDE	Free	Many	C	3D	C	GUI	http://amide.sourceforge.net/
Amira	Demo	Many	Java	3D	No	GUI	http://amira.zib.de/
Anatomist/BrainVISA	Free	Many	C++	Image	C++	?	http://brainvisa.info/
Apteryx	-	Many	Java	?	?	?	http://imaging.apteryx.fr/dicom/
Blaze	Demo	?	?	?	?	GUI	http://www.absoluteshareware.com/most/dcm/
C code for DICOM	Offline	-	C	-	-	-	http://www.rad.upenn.edu/~grevera/
CarDiCon	Free	Win	-	Image	No	GUI	http://fhs-consulting.com/cardicon/E_cardicon.html
Casimir	-	Win	-	Network	No	Exe	http://www.sim.hcuge.ch/casimir/01_CASIMIR_Presentation_FR.htm
CDMEDIC	-	Mac/Unix		PACS			http://sourceforge.net/project/showfiles.php?group_id=61662
CHILI Digital Radiology	$			Misc			http://www.chili-radiology.com/en/solutions/
CImageBufferX	Demo					ActiveX	See Puntoexe
ComView	Offline	-	-	-	-	-	-
Conquest	Free	Win/Unix	C++	MW	C++	Console	http://www.xs4all.nl/~ingenium/dicom.html
CoreWare	Free	-	-	-	-	-	http://www.coreware.co.kr/
CTN							http://erl.wustl.edu/research/dicom/ctn.html
CTN - Mallinkrodt							ftp://wuerlim.wustl.edu/pub/dicom/software/ctn/
dcm2nii	-			Image			http://www.sph.sc.edu/comd/rorden/mricron/dcm2nii.html
dcm4che	-	Many	Java				http://www.dcm4che.org/
dcm4che SourceForge	-	Many	Java				http://sourceforge.net/projects/dcm4che/
DCMViewer	Offline	-	-	-	-	-	http://www.fruitfruit.com/vc/
Dejarnette	Offline						http://www.dejarnette.com/1024/splashpage.html
DesAcc	Website	-	-	-	-	-	http://www.desacc.com/
DICOM Anonymizer Light	-	Win	Java				http://eng.neologica.it/download/downloadDICOMAnonymizer.html
Dicom Calculator 4.0				?			http://www.softwareindex.org/Dicom-downloads_23379.htm
DICOM CareOn	Offline	-	-	-	-	-	-
DICOM Compare							See TomoVision
DICOM Detective	Demo						http://www.dicomtool.com/
DICOM Dumper	-	Win	Java				http://eng.neologica.it/download/downloadDICOMDumper.html
DICOM for PowerPoint	-			Image			http://radfiler.com/dicomppt.htm
DICOM Gateway	$			Misc			http://www.encomate.com.tw/products_dicomgateway.php
DICOM Inspector.NET				?			http://www.softpedia.com/get/Others/Miscellaneous/DICOM-Inspector-NET.shtml
DICOM parser							http://www.codeproject.com/KB/cpp/DICOM_parser.aspx
DICOM Print Manager	-			Print			See HIPAX
DICOM Printer	Demo						See Flux
DICOM Solutions	Website	-	-	-	-	-	http://www.dicom-solutions.com/index.shtm?lang=en
DICOM ViewerMedImgTool				?			http://dicom-viewer-medical-image-tool.http-www-eplanetlabs-net.qarchive.org/
DICOM ViewStar PC	Offline	Win	-	-	-	-	http://www.charrua.com/ingenio/cardiovista/indexen.htm
DICOM win 32 DLL	$						See Flux

Name	Cost	Platform	Language	Type	Source	Category	URL
DICOM Worklist Server	-			MW			See HIPAX
DICOM#							http://sourceforge.net/projects/dicom-cs/
dicom2	Free	Win/Unix	C++	Files	No	Console	http://barre.nom.fr/medical/dicom2/
Dicom2Pict		Mac					ftp://ftp.funet.fi//pub/mac/info-mac/gst/grf/dicom-2-pict.hqx
Dicom3tools							http://www.dclunie.com/dicom3tools.html
DICOM4india	Offline	-	-	-	-	-	http://www.websamba.com/dicom4india
dicom4j	-	Many	Java	Toolkit	Yes	Lib	http://dicom4j.sourceforge.net/index.html
DICOMaccess	Demo						See DesAcc
DICOMatic							See TomoVision
DICOMBMP	Offline	Win	-	-	-	-	ftp://ftp.xray.hmc.psu.edu/dicom_software/dicombmp.zip
DICOMgateway	Demo						See DesAcc
DICOMLIB		Win/Unix	C++				http://dicomlib.swri.ca/joomla/index.php?option=com_frontpage&Itemid=1
DicomLite		Win					http://www.3d-head.com/
DicomObjects			Many			ActiveX	http://www.medicalconnections.co.uk/html/dicomobjects.html
DicomPshop	-	Mac		Image			http://www.dclunie.com/dicom-plugin/
DICOMscope	Offline	Win	Java	-	-	-	http://www.offis.de/projekte/dicom/soft-docs/soft03_e.html
DicomStar							See Dicom Solutions
dicom-tcpdump	-			Network			http://www.tcpdump.org/
DicomWorks (Inviweb)		Win					http://dicom.online.fr/
DicomXBridge	Demo				No	Lib	See Puntoexe
Digital Jacket	Demo						See DesAcc
Dr Razz	-	Mac					http://www.dr-razz.com/home.html
dump2dcm	-			Misc			http://pwet.fr/man/linux/commandes/dump2dcm
DVS	Free	Win	Exe		No		http://www.ee.bilkent.edu.tr/~cetin/vs.html
DVTk	Free	Win	C#	Files	C#	GUI/DLL	http://dicom.dvtk.org/index.php
Easymage	Offline	-	-	-	-	-	http://www.aetnet.it/products/Easymage
Escape							http://www.escape.gr/dicom/index.html
ETIAM (DicomEye)	$						http://www.etiam.fr/en/maj-e/c2a1i894b1/etiam.htm
Eviewbox	-	Many	Java				http://eviewbox.sourceforge.net/
ezDICOM		Win					http://www.sph.sc.edu/comd/rorden/ezdicom.html
Floating Point Systems	Offline	-	-	-	-	-	http://www.floating.co.uk/idl/dicom.html
Flux	Website						http://fluxinc.ca/products.html
FP Image	Demo	Win					http://www.fpimage.com/info.html
GAMMA	Free	Win					http://www.uni-mainz.de/FB/Medizin/Nuklearmedizin/Physik/Gamma_PC2.html
Gdcm							http://www.creatis.insa-lyon.fr/Public/Gdcm/
GMATEK							http://pagesperso-orange.fr/gmatek/index.htm
GraphicConverter		Mac					http://www.lemkesoft.com/
Health Data Explorer	Demo						See DesAcc
HIPAX	-	Win		Misc			http://www.hipax.de/pageseng/products/product.htm
IDICON				?			http://www.inf.u-szeged.hu/~idicon/
IfranView Plugins	-			Image			http://www.google.com/search?q=irfanview+dicom
ImageJ	-	Many	Java				http://rsbweb.nih.gov/ij/
ImageMagick	-	Many	Many	Image	Yes	Lib	http://www.imagemagick.org/script/index.php
ImageNet	Offline	Win	-	-	-	-	http://anchorage.umaryland.edu/imagenet.html
ImageServer	Free	Many	Java	Multi	Yes	Archi	http://www.mii.ucla.edu/index.php/MainSite:ImageServerHome
Imebra			C++	Files	Yes	Lib	See Puntoexe
Imread	-	Win	Java				http://www.uchsc.edu/sm/neuroimaging/download/imread/imread1.htm
INFINITT	$			PACS			http://www.mediface.com/eng/solution/solution_main.asp?pageNum=1

InteleViewer	Demo						http://www.intelerad.com/en/download.php?pid=2&loc=iw
InterFormat	Offline	-	-	-	-	-	http://www.radio-logic.com/rliwebpg-Download.html
iRad	-	Mac					http://irad.sourceforge.net/
Java Dicom Toolkit	-	Many	Java				http://www.trispark.com/
JAVA Viewer	-	Many	Java				http://mars.elcom.nitech.ac.jp/dicom/index-e.html
JDCM	-	Many	Java				http://www.geocities.com/gigiobb/
JDicom	Offline	Many	Java	-	-	-	http://www.tiani.com/JDicom/
JDicomViewer	Offline	Many	Java	-	-	-	http://wwwusers.imaginet.fr/~sderhy/dicJava.html
JiveX	-	Many	Java				http://www.visus-tt.com/download/dicom-viewer.html
JPEG-LS	-			Image			http://dclunie.com/jpegls.html
Julius	Offline	Win/Unix	-	-	-	-	http://julius.caesar.de/
Laurel Bridge Software	$	Many					http://www.laurelbridge.com/home.php
LeadTools	Demo						http://www.leadtools.com/SDK/Medical/Medical-DICOM.htm
Linomed	Offline	-	-	-	-	-	http://www.linomed.de/
LONI	Free	Many	Java			GUI	http://www.loni.ucla.edu/Software/Software_Detail.jsp?software_id=54
MacAngioView	Offline	Mac	-	-	-	-	http://home.attbi.com/~wlapointe/
Madena	-	Mac					http://radonc.usc.edu/USCRadOnc/Madena/Madena.html
MathWorks	$						http://www.mathworks.com/products/image/
MEDAL	Offline	-	-	-	-	-	http://sonic.image.med.osaka-u.ac.jp/reza/medal.html
Medasys (DxWin)	Offline	-	-	-	-	-	http://www.medasys-digital-systems.fr/medical/dxwin.html
MedicView	Demo			?			http://www.medicimaging.com/
Medigration	-			PACS			http://www.medigration.de/
MedImaView		Win				GUI	See Dicom Solutions
MedINRIA							http://www-sop.inria.fr/asclepios/software/MedINRIA/
MEDISP		Win				Exe	http://www.teiath.gr/stef/tio/medisp/downloadMEDISPDICOMViewer.htm
MedVision	Offline	Mac	-	-	-	-	http://www.imnet.com/
Medx	Offline	-	-	-	-	-	http://topaz.sensor.com/links/mg/props/dicom
Merge (eFilm)	Demo						http://www.merge.com/hdw/
MergeCOM-3	Demo						https://www.merge.com/emea/estore/content.aspx?productID=122
MESA	-			Misc			http://wuerlim.wustl.edu/mesa/
MeVisLab	Demo	Many					http://www.mevislab.de/
MillenSys	Demo	Win					http://www.millensys.com/
MIPAV	-	Many	Java				http://mipav.cit.nih.gov/
MITA	Offline	-	-	-	-	-	http://www.voicenet.com/~mita
MRIConvert							http://lcni.uoregon.edu/~jolinda/MRIConvert/
MRIcro		Many					http://www.sph.sc.edu/comd/rorden/mricro.html
MRIcron		Many		?			http://www.sph.sc.edu/comd/rorden/mricron/index.html
MultiTech msiCOM3	Waiting	Win/Unix	COM.C++				http://www.mulsol.com/toolkits.htm
My DICOM	$						http://www.mydicom.net/CSharp.aspx
MyPACS.net	Offline			PACS			http://www.mypacs.net/repos/mpv3_repo/static/m/Solutions/DAR.htm
NeatVision (ex-NeatMed)	-	Many	Java				http://www.vsg.dcu.ie/code.html
NeuroModeller	Offline	Win	-	-	-	-	http://users.infohouse.com/amiller/home.htm
NeuroModeller	Offline	Win					http://www.broadviewnet.net/amiller/
NIH Image	Offline	-	-	-	-	-	ftp://zippy.nimh.nih.gov/pub/nih-image/documents/
OFFIS							http://dicom.offis.uni-oldenburg.de/index.php.en
OFFIS FTP							ftp://dicom.offis.uni-oldenburg.de/pub/dicom/offis/software
OpenDICOM.NET							http://opendicom.sourceforge.net/
OpenSourcePACS	Free	Many	Java	PACS			http://www.mii.ucla.edu/index.php/MainSite:OpenSourcePacsHome

OpenSourcePACS	Free	Many	Java		Yes		http://www.mii.ucla.edu/index.php/MainSite:OpenSourcePacsHome
Osiris							http://www.sim.hcuge.ch/osiris/01_Osiris_Presentation_EN.htm
OsiriX	-	Mac					http://www.osirix-viewer.com/
Oviyam	-			Misc			http://skshospital.net/oviyam/
PacsOne Server	-			PACS			http://www.pacsone.net/
PACSview (ACRNview)	Offline	-	-	-	-	-	http://www-ang.kfunigraz.ac.at/~zeilinge/pacsview/PACSview.html
Papyrus	GNU	Many	C				http://www.sim.hcuge.ch/papyrus/01_Papyrus_Presentation_EN.htm
PET-CT Viewer	Free	Win					http://bidmc.harvard.edu/content/bidmc/Departments/Radiology/nm_download.asp
Philips							http://www.medical.philips.com/main/company/connectivity/
Philips DICOM Viewer							http://netforum.medical.philips.com/movie/PmsDView.exe
PixelMed	-	Many	Java				http://www.pixelmed.com/index.html#PixelMedJavaDICOMToolkit
Planar (Dimplx)	$						http://www.planar.com/support/Support_By_Product/Medical/archive/
PowerDicom							See Dicom Solutions
Puntoexe	Website	-	-	-	-	-	http://puntoexe.com/
PVRG JPEG	-			Image			http://dclunie.com/jpegge.html
Radiological Analysis Tool							http://members.tripod.com/tonyrudd/
RadPix	Demo						http://www.radpix.com/
RadXtreme	Demo						http://www.radxtreme.com/radxt.html
RAIM	Offline	Win					http://www.cspt.es/webcspt/udiat/labimg/labimg.htm
Reliable Medical DICOM.net	Demo						http://www.reliable-med.com/Products.aspx
reViewMD	Offline	-	-	-	-	-	http://www.digisoftdirect.com/Products/reviewmd.html
Rubo Medical Imaging		Win					http://www.rubomedical.com/index.html
Santesoft							http://users.forthnet.gr/ath/mkanell/
SimpleDICOM	Free	Win					http://www.healthimaging.com/content/view/8721/160/
SiteView	-	-	-	-	-	-	ftp://wuarchive.wustl.edu/graphics/graphics/demo-packages/SiteView/
Skyline Delphi	-			Misc			http://skylinetools.com/imagelib/index.html
sliceOMatic							See TomoVision
Slicer		Many			Yes		http://www.slicer.org/
Snowbound	$						http://www.snowbound.com/format/dicom.html
SRView	-	Unix					http://www.synergyresearch.com/SRView.html
TagViewer	Offline	-	-	-	-	-	http://www.geocities.com/SiliconValley/Chip/3967/dicom.htm
TDICOMCAD	Demo						http://www.codeidea.com/html/components.html
TIMS	Demo						http://www.tims.com/
TomoMagine	Free	Win				Exe	http://tomomagine.de/index2.htm
TomoVision							http://www.tomovision.com/
TrueView	Offline	-	-	-	-	-	http://www.truevcl.com/
UC Davis Medical Center	Offline	-	-	-	-	-	http://imrad.ucdmc.ucdavis.edu/
UniPACS	Offline	-	-	-	-	-	http://www.unipacs.com/
UnLimiter	Offline	Win					http://www.unlimiter.co.uk/html/viewer.htm
Varian	$			Misc			http://www.varian.com/
ViewPlus							http://www.sandersdata.com/ViewPlus.htm
Visage PACS				PACS			http://www.visageimaging.com/products/visage_pacs.php
VisPoint	Offline	-	-	-	-	-	http://members.aol.com/VisPoint/
VTK	Free		C++	Image	Yes		http://public.kitware.com/VTK/
XMedCon							http://xmedcon.sourceforge.net/Main/HomePage
XNView	Free	Many					http://www.xnview.com/

APPENDIX E: DICOM FILES SAMPLES SOURCES

Name	Source Location
3DVIEWNIX	http://www.mipg.upenn.edu/~Vnews/3dviewnix_ftp.html#data
AAPM	http://deckard.mc.duke.edu/~samei/tg18#_DOWNLOAD_THE_TG18
AGFA	http://www.agfa.com/en/he/support/doc_library/dicom/adc_dicom_images/index.jsp
AGFA FTP	http://ftp.agfa.com/pub/med/zip/
AMIDE	http://amide.sourceforge.net/samples.html
D. Clunie	http://www.dclunie.com/images/
Dicom.net	http://www.reliable-med.com/Download/DicomNet.ExampleImages.zip
Gdcm	http://www.creatis.insa-lyon.fr/~jpr/PUBLIC/gdcm
LeadTools	http://www.leadtools.com/SDK/Medical/DICOM/ltdc19.htm
Merge	https://www.merge.com/EMEA/helpcenter/onlinesupport_sampleimages.asp
Micropos	Provided by the company
OASIS	http://www.oasis-brains.org/app/template/Tools.vm
OFFIS	ftp://dicom.offis.uni-oldenburg.de/pub/dicom/images/
OsiriX	http://pubimage.hcuge.ch:8080/
RadPix	http://www.radpix.com/download.asp
RSNA	http://erl.wustl.edu/research/dicom/imageindex.html
RSNA FTP	ftp://ftp.erl.wustl.edu/pub/dicom/images/version3/
Rubomed	http://www.rubomed.com/dicomfiles/index.html
S. Barré	http://www.barre.nom.fr/medical/samples/
srdoc103	ftp://dicom.offis.de/pub/dicom/offis/software/dscope/dscope360/support/srdoc103.zip
Various	http://apps.sourceforge.net/mediawiki/gdcm/index.php?title=Sample_DataSet
Web Mathematica	http://www.wolfram.com/products/webmathematica/resources/dicom.html