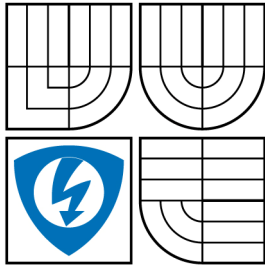


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKACNÍCH
TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ**

**FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS**

ANALÝZA PROTOKOLU DICOM A JEHO VYUŽITÍ PRO ZPRACOVÁNÍ OBRAZOVÝCH DAT

ANALYSIS PROTOCOL DICOM AND HIS USAGE FOR PROCESSING IMAGE DATA

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

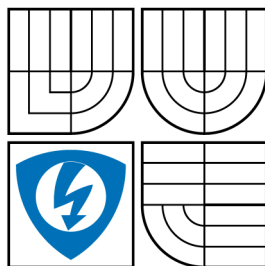
AUTOR PRÁCE
AUTHOR

LUKÁŠ VALOUCH

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. ONDŘEJ ŠMIRG

BRNO 2009



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: Lukáš Valouch

ID: 72131

Ročník: 3

Akademický rok: 2008/2009

NÁZEV TÉMATU:

Analýza protokolu DICOM a jeho využití pro zpracování obrazových dat

POKYNY PRO VYPRACOVÁNÍ:

DICOM je protokol používaný v lékařství pro ukládání a přenos dat z lékařských přístrojů. Cíl bakalářské práce spočívá v rozboru tohoto standardu se zaměřením na přenos a zpracování obrazových dat. Výsledkem bude aplikace vytvořená v C,C++ schopná číst a zapisovat obrazová data v rámci DICOM protokolu a dále s těmito daty dělat základní úpravy (např. úprava jasu).

DOPORUCENÁ LITERATURA:

- [1] MATOUŠEK L.: Waveletová analýza a zvýrazňování MR tomografických a ultrazvukových obrazů, Diplomová práce, Brno: VUT, Fakulta elektrotechniky a komunikačních technologií, 2008, 52s.
[2] ŘÍHA, K.: Pokročilé techniky zpracování obrazu. Elektronické texty VUT, Ústav telekomunikací FEKT VUT v Brně, 2007

Termín zadání: 9.2.2009

Termín odevzdání: 2.6.2009

Vedoucí práce: Ing. Ondřej Šmirg

prof. Ing. Kamil Vrba, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

Abstrakt

Tato bakalářská práce popisuje DICOM, jakožto souhrn standardů pro zpracování, ukládání a přenos medicínských obrazových dat. Standard je vyvíjen organizacemi ACR a NEMA, je značně rozsáhlý, jeho aktuální verze z roku 2008 se skládá z 18 dokumentů. Pro zpracování obrazových dat je důležitá část samotného formátu dat a souborů, tzv. off-line komunikace.

Následuje popis aplikace implementované v jazycích C, C++ pomocí knihovny Imebra, která je k dispozici zdarma pod licencí GNU GPL. Výsledný program vychází tedy ze standardu DICOM, který je rozebrán výše. Zabývá se problémy zobrazování obrazů 16 bitové hloubky na běžných zobrazovacích jednotkách o 8 bitech, jako jsou např. monitory počítačů. Aplikací je možno zobrazovat DICOM obrazy pouze po „vrstvách“, jednotlivých separovaných bitech podle úrovně důležitosti.

Klíčová slova

DICOM, Imebra, C++, struktura, obrazová data, formát, CLR.

Abstract

This Bachelor thesis describes DICOM which is a summary of standards for handling, storing and transmitting information in medical imaging. This standard is developed by organizations ACR and NEMA, it is very large file and it contains 18 of documents which are an actual version of this norm. The last norm was got out in 2008. The most important part of this format of date and files is data image processing so called off-line communication.

It follows description of implemented application in C, C++ languages by the help of Imebra library. This Imebra library is free in terms of license of GNU GPL. The resultant program is standard of DICOM from, which is described above. This program is concerned with imaging of 16 bits of dept on 8 bits of dept of display devices, for example common computer displays. This application is able to display several layers of DICOM's pictures. Each of layers contains separated bites which are presented by importance of levels.

Keywords

DICOM, Imebra, C++, structure, image data, format, CLR.

VALOUCH, L. *Analýza protokolu DICOM a jeho využití pro zpracování obrazových dat*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 42 s. Vedoucí bakalářské práce Ing. Ondřej Šmirg.

Prohlášení

Prohlašuji, že svou bakalářskou práci na téma „Analýza protokolu DICOM a jeho využití pro zpracování obrazových dat“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne

.....

podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Ondřeji Šmirgovi za velmi užitečnou metodickou pomoc a cenné rady při zpracování práce.

Obsah

Úvod	9
1 Standard DICOM	10
1.1 Vznik DICOMu a jeho charakteristika	10
1.2 Základní funkční principy	11
1.2.1 Aplikační entita, informační objekt, servisní třída, SOP třída	11
1.2.2 Hlavní komunikační model	12
1.3 Model úložného prostoru	13
1.3.1 Vrstva fyzických médií (Physical Media Layer)	13
1.3.2 Vrstva formátu médií (Media Format Layer)	14
1.3.3 Vrstva formátu dat (Data Format Layer)	14
1.3.4 DICOM aplikační profily úložného prostoru (Media Storage Application Profiles) ...	15
1.3.5 Úložný prostor a DICOM standardní struktura	15
1.4 Formát DICOM souboru	16
1.4.1 Meta informace DICOM souboru	16
1.4.2 Zapouzdření datové sady	17
1.4.3 Podpora práce se soubory	17
1.4.4 Zabezpečený formát DICOM souboru	17
1.5 Služby DICOM souboru (DICOM File Service)	18
1.5.1 Souborová sada (File set)	18
1.5.2 Souborové ID	19
1.5.3 Funkce a služby správy souboru (File Management Roles and Services)	19
1.5.4 Znaková sada	19
1.5.5 Rezervovaný ID souboru DICOMDIR (Reserved DICOMDIR File ID)	20
1.6 Kódování dat	20
1.6.1 Podpora znakové sady pro grafické znaky	20
1.6.2 Datová reprezentace (Value Representation, zkráceně VR)	21
1.6.3 Datová vícenásobnost (Value multiplicity, zkráceně VM) a ohraničení	21
1.7 Datová sada (Data Set)	22
1.7.1 Prvky dat	22
1.7.2 Délka skupiny (Group Length)	24
1.7.3 Řazení bajtů (Byte Ordering), velký a malý endian	24
1.7.4 Typ datového prvku	25
1.7.5 Vnořené datové sady	25
1.7.6 Privátní datové prvky (Private data elements)	26
1.8 Zakódování datových prvků: Pixel, Overlay a Waveform data	27
1.8.1 Zakódování Pixel dat ze souvisejících datových prvků	27
1.8.2 Zakódování Overall dat ze souvisejících datových prvků	28
1.8.3 Nativní nebo zapouzdřený formát zakódování	28
1.9 Unikátní identifikátory (Unique Identifiers, zkráceně UID)	28
1.10 Přenosová syntaxe	29
2 Popis výsledného programu	30
2.1 Imebra	30
2.2 Přidání knihovny Imebra do projektu	30

2.3	Uložené informace.....	31
2.4	Struktura programu.....	31
2.5	Vzhled aplikace.....	32
3	Výsledky práce.....	34
4	Závěr.....	38
	Literatura.....	39
	Přílohy.....	41

Seznam obrázků

Obr. 1.1	Hlavní komunikační model DICOM.....	12
Obr. 1.2	Model úložného prostoru.....	13
Obr. 1.3	Úložný prostor a DICOM části.....	16
Obr. 1.4	Formát souborů.....	16
Obr. 1.5	Uspořádání DICOM datové sady a datových prvků.....	22
Obr. 2.1	Příklad uložení pixelu.....	31
Obr. 2.2	Vývojový diagram metody zobraz.....	32
Obr. 2.3	Nastavení parametrů.....	32
Obr. 2.4	Celkový vzhled aplikace (upraveno, aby nebyly obsaženy osobní informace pacienta).....	33
Obr. 3.1	Uložené obrazové bity.....	34
Obr. 3.2	Prvních 8 bitů MSB, MONOCHROME2.....	34
Obr. 3.3	Prvních 8 bitů MSB, MONOCHROME1.....	34
Obr. 3.4	Rozklad do bitových hladin (vlevo nahoře bity na 11 pozici, vpravo dole na 0 pozici).....	35
Obr. 3.5	Posun „okénka“ vybíraných bitů.....	35
Obr. 3.6	RGB, prvních 8 MSB.....	36
Obr. 3.7	RGB, vrstvy 8 – 0.....	36
Obr. 3.8	MON_2, 11b, MSB barva R=3.....	36
Obr. 3.9	MON_2, 11b, MSB barva R=7.....	36
Obr. 3.10	Rozdělení bitů do 2 barev (vlevo znázorněn Obr. 3.8, vpravo Obr. 3.9).....	37
Obr. 3.11	MON_2, 11b, zaokrouhlování.....	37
Obr. 3.12	MON_2, 11b, každý druhý bit.....	37
Obr. 3.13	Přepočítávání vzorků (vlevo zaokrouhlením, vpravo každý druhý bit).....	37

Seznam tabulek

Tab. 1.1	Analogie vytváření věty a SOP třídy.....	11
Tab. 1.2	Výňatek z datových reprezentací.....	21
Tab. 1.3	Datový prvek s explicitní VR.....	23
Tab. 1.4	Datový prvek s jinou explicitní VR než je v Tab. 1.3.....	24
Tab. 1.5	Datový prvek s implicitní VR.....	24
Tab. 1.6	Příklad datového prvku s implicitní VR SQ s 3 položkami explicitní délky.....	26
Tab. 1.7	Příklad datového prvku s explicitní VR SQ nedefinované délky, obsahující 2 položky explicitní délky.....	26

Seznam symbolů, veličin a zkratek

Zkratky

ACR	American College of Radiology – Americká univerzita radiologie
AE	Application Entity – aplikační entita
ANSI	American National Standards Institute – Americký normalizační institut
API	Application Programming Interface – rozhraní aplikačních programů
AT	Attribute Tag – datová reprezentace
CD-R	Compact Disk-Recordable – optický datový nosič s možností zápisu
CEN	European Committee for Standardization – Evropský výbor pro normalizaci
CLR	Common Language Runtime – grafická nástavba Visual Studia 2005
CR	Carriage Return – řídicí znak značící nový řádek
CT	Computer Tomography – počítačová tomografie
DICOM	Digital Imaging and Communications in Medicine – standard pro zobrazování a komunikaci v medicíně
DIMSE	DICOM Message Service Element- služby výměny zpráv
ESC	Escape - řídicí znak značící ukončit
FAT	File Allocation Table – tabulka souborového systému
FD	Floating Point Double – datová reprezentace
FF	Form Feed - řídicí znak značící kód pro posun o stránku
FL	Floating Point Single – datová reprezentace
FSC	File-set Creator – vytvoření souborové sady
FSR	File-set Reader – čtení souborové sady
FSU	File-set Updater – aktualizování souborové sady
HL7	Health Level Seven – organizace podílející se na vývoji mezinárodních zdravotnických norem
ID	Identifier – identifikátor
IEEE	Institute of Electrical and Electronics Engineers – jedna z předních standardizačně vývojových organizací na světě
IOD	Information Object Definition – informační objekt
ISO	International Standardizing Organization – Mezinárodní normalizační organizace
JIRA	Japan Industries Association of Radiological Systems – Asociace japonských výrobců radiologických systémů
JPEG	Joint Photographic Experts Group – kompresní mechanismus
JPIP	JPEG 2000 Interactive Protocol – kompresní přenosový protokol

LF	Line Feed – řídicí znak značící posun řádků
LSB	Least Significant Bit – nejméně významný bit
MOD	Magneto Optical Disk – magneto-optický datový nosič
MPEG	Motion Pictures Experts Group – kompresní mechanismus
MR	Magnetic Resonance – magnetická rezonance
MSB	Most Significant Bit – nejvíce významný bit
NEMA	National Electronics Manufactures Association – Národní asociace výrobců elektrotechniky
NIU	Network Interface Unit – zařízení síťového rozhraní
OB	Other Byte String – datová reprezentace
OF	Other Float Sting – datová reprezentace
OSS	Open Source Software, nebo jen Open Source – počítačový software s otevřeným zdrojovým kódem, dostupný technicky i licenčně
OW	Other Word String – datová reprezentace
PACS	Picture Archiving and Communication Systems – systém pro obrazovou archivaci a komunikaci
RFC	Request For Comments – označení řady standardů popisujících protokoly
RLE	Run Length Encoding – kompresní mechanismus
RTTI	Run Time Type Identification – dynamická identifikace typu
SL	Signed Long – datová reprezentace
SOP	Service Object Pair – dvojice servisních objektů
SQ	Sequence of Items – datová reprezentace
SS	Signed Short – datová reprezentace
TCP/IP.	Transmission Control Protocol / Internet Protocol – sada síťových komunikačních protokolů
UI	Unique Identifier – datová reprezentace
UID	Unique Identifier – unikátní identifikátor
UL	Unsigned Long – datová reprezentace
UN	Unknown – datová reprezentace
US	Unsigned Short – datová reprezentace
UT	Unlimited Text – datová reprezentace
VM	Value Multiplicity – vícenásobná data
VR	Value Representation – datová reprezentace

Úvod

Náš život nám mohou značně znepríjemnit nemoci. V těchto chvílích vyžadujeme maximální možné ošetření ze strany lékařů a využití nejmodernější medicíny. Nemalou roli při zlepšování péče o pacienta jsou vyšetření pomocí digitálních zobrazovacích zařízení, v dnešní době to vnímáme jako běžnou záležitost. Aby se mohli jednotlivé technologie snáze rozšiřovat, bylo zapotřebí jisté standardizace.

První část bakalářská práce si klade za cíl přiblížení DICOM (Digital Imaging and Communications in Medicine), jakožto souhrnu standardů pro zpracování, ukládání a přenos medicínských obrazových dat. Je to veřejný a obecně hojně používaný standard v medicíně, který je vyvíjený organizacemi ACR a NEMA. DICOM se skládá z více dokumentů. V současné době je v aktuální verzi z roku 2008 celkem 18 částí, z nichž 2 jsou označeny jako vysloužilé (viz. příloha A). Dalo by se říct, že se tento standard zabývá dvěma hlavními oblastmi:

- protokolem pro přenos obrazových dat,
- formátem obrazových dat.

Komunikace pomocí DICOMu mezi jednotlivými lékařskými přístroji a počítači je už vcelku běžná. V České republice se již začíná využívat výměny dat na základě tohoto protokolu v globálnějším měřítku mezi jednotlivými nemocnicemi [13]. Pro druhou část práce je důležitější část samotného formátu dat a souborů, tzv. off-line média.

Běžné zobrazovací jednotky, jako např. monitory počítačů, dokáží pracovat pouze s 8 bity hloubky barev. U medicínských přístrojů, kde jsou vyžadovány větší detaily, bývají zobrazovače kvalitnější. Obrazové informace v DICOM souborech mohou být ukládány v různých bitových hloubkách, u lékařských přístrojů většinou v 12 bitech. Hlavním problémem většiny volně šiřitelných programů, které čtou DICOM obrazové soubory s větší bitovou hloubkou než 8 je, že při zobrazení vezmou pouze 8 bitů s nejdůležitějšími informacemi (MSB) a zbylé méně významné odstraní. Dochází tím však ke ztrátě informace, která je při detailních studiích podstatná.

Výsledná aplikace se snaží umožnit pozorování těchto odstraňovaných bitů s nejnižší informační hodnotou na 8 bitových zobrazovačích. Využívá k tomu práci s jednotlivými bity pixelů, které si v rámci celého snímku můžeme představit jako jeho řezy. Program umožňuje:

- zobrazovat snímky ve 4 barevných modelech,
- volitelně vybírat bitové hladiny,
- rozdělit bitové hladiny na dvě části pomocí dvou základních barev modelu RGB,
- nastavovat způsob přepočítávání velkých jasových hodnot.

Z důvodu ochrany osobních informací jsou v práci uvedené náhledy grafického uživatelského rozhraní náležitě upraveny a studijní DICOM soubor není uložen na příkládaném CD.

1 Standard DICOM

1.1 Vznik DICOMu a jeho charakteristika

V 80. letech minulého století přišel rozvoj počítačové tomografie – CT, digitalizace obrazových dokumentů a s tím související zvýšení používání počítačů v klinických aplikacích. Začala se objevovat potřeba standardizované metody převodu obrazu a doprovodných informací mezi medicínskými zařízeními vyráběnými různými firmami. Proto společným úsilím Americké vysoké školy radiologické (ACR) a Národní asociace výrobců elektroniky (NEMA) vznikl v roce 1983 společný výbor, který vytvořil standardy k:

- podpoře komunikace digitální obrazové informace, bez ohledu na výrobce přístroje,
- usnadnění vývoje a expanze obrazových archivačních a komunikačních systémů PACS s podporou rozhraní v nemocničních informačních systémech,
- vytváření databází diagnostických informací, které mohou být využity širokou škálou geograficky distribuovaných zařízení.

Výsledkem byl ACR-NEMA Standards Publication 300-1985 verze 1.0, publikovaný v roce 1985. Následovaly dvě opravy: první datovaná k říjnu 1986 a druhá k lednu 1988.

V roce 1988 byla vydána verze 2.0, která zahrnovala verzi 1.0 s publikovanými a dodatečnými opravami. Současně obsahovala nový materiál pro poskytnutí řízení podpory zobrazovacích jednotek, k čemuž byl představen nový hierarchický návrh k rozpoznávání obrazu a přidávání datových elementů pro zvětšení přesnosti při popisu obrazu. Předepisoval hardwarové rozhraní, minimální softwarovou sadu příkazů a shodný souborový formát dat.

V roce 1992, během vývoje třetí verze, dostal standard jméno DICOM, aby zlepšil možnost mezinárodního přijetí jako standardu. Byly definované nové servisní třídy a přidána síťová podpora. Poslední verze standardu je formálně ještě 3.0, stále aktualizovaná a prodlužovaná od roku 1992. Namísto používání čísla verze standardu je často uváděn rok vydání, jako „the 2008 version of DICOM“.

Standard je vyvíjen ve spojení s dalšími standardizačními organizacemi zahrnujícími CEN TC251 v Evropě a JIRA v Japonsku, včetně dalších organizací IEEE, HL7 a ANSI v USA.

DICOM je vhodný pro síťové prostředí, starší verze byly použitelné jen v sítích typu bod-bod, byl nutný NIU (Network Interface Unit). DICOM již podporuje operace v síťovém prostředí pomocí standardního síťového protokolu TCP/IP.

Starší verze nespécifikovala formát souboru, volbu fyzických prostředků nebo logický souborový systém. DICOM podporuje operace v off-line mediálním prostředí, používající průmyslových prostředků jako CD-R a MOD a logické souborové systémy jako ISO 9660 a PC systémy souborů (FAT16).

Starý ACR-NEMA byl omezen v přenosu dat, ale DICOM je specifikovanější, přes koncept servisních tříd, sémantiku příkazů až k přidruženým datům. DICOM explicitně popisuje, jak musí realizátor uspořádat úrovně shody (Conformance Statement) se standardem DICOM k tomu, aby vybral jeho specifické možnosti.

DICOM je strukturovaný jako vícedílný dokument (po vzoru ISO směrnic), což usnadňuje jeho vývoj v rychle se rozvíjejícím prostředí, zjednodušuje to přidávání nových vlastností. Každý dokument tohoto standardu je identifikován názvem a číslem, které je ve formě "PS 3.X-YYYY", kde číslo 3 určuje verzi, X je obecně označováno jako číslo standardu a YYYY je rok (např. DICOM část

1 je označena názvem „Introduction and Overview“ a číslo dokumentu je PS 3.1-2008). Pro zjednodušení budu v následujícím textu používat pouze označení PS 3.X, jelikož se budu zabývat především aktuální verzí z roku 2008.

DICOM představuje explicitní informační objekty nejen pro obrazy a grafiku ale také pro křivky, zprávy, tisk, atd. Specifikuje stanovenou techniku pro jedinečně se identifikující každý informační objekt, jež komunikuje napříč sítí.

Rozvíjející standard je udržovaný v souladu s DICOM standardizující komisí (Procedures DICOM Standards Committee). Návrhy pro zlepšení jsou chystané od DICOM Committee členských organizací založených na vstupu od uživatelů standardu. Tyto návrhy jsou zvažované pro zahrnutí v budoucím vydání tak, aby byla podporována slučitelnost s předchozími vydáními.

V rámci rozvoje se také označují vysloužilé nepoužívané části (retired). Není přímo zakázáno je používat, ale není již udržována dokumentace k těmto částem. Je obsažena pouze v předchozích verzích a jejich užití v nových implementacích není schvalováno. [1]

1.2 Základní funkční principy

1.2.1 Aplikační entita, informační objekt, servisní třída, SOP třída

Pro pochopení funkční úrovně DICOMu je dobrá znalost několika základních prvků:

- aplikační entita – aplikace pracující s DICOM daty (Application Entity, zkráceně AE)
- informační objekt – definuje jádro obsahu medicínského snímání
- servisní třída – definuje co se má udělat s obsahem snímání

Informační objekt (Information Object Definition, zkráceně IOD) [2] je abstraktní objektový model, který popisuje pomocí atributů vlastnosti skutečného objektu (Real World Object, např. pacient). Má dvě třídy:

- normalizovanou – zahrnující jen atributy aktuální entity,
- kompozitní – obsahuje i atributy přidružené.

Např. kompozitní třída CT obrazů zahrnuje nejen atributy v obraze přímo obsažené (denní datum, času), ale i atributy které v obraze nejsou (jméno pacienta).

Ze servisní třídy (Service Class) [3] vychází služby výměny zpráv (DICOM Message Service Element, zkráceně DIMSE) [6]. Tento soubor pravidel umožňuje mezi jednotlivými AE služby výměny zpráv.

Kombinací IOD a DIMSE nám vzniknou funkcionální jednotky DICOMu nazvané dvojice servisních objektů (Service Object Pair, zkráceně SOP), od ustanovení standardu označovaných jako SOP třída. Všechno co se v DICOMu dělá je založeno na použití SOP tříd. Vytváření takové třídy je víceméně analogií k vytváření věty:

Tab. 1.1 Analogie vytváření věty a SOP třídy

sloveso:	ukládání	→	služby výměny zpráv (DIMSE)
podstatné jméno:	CT obrázek	→	informační objekt
obecná věta:	ukládání CT obrazů	→	SOP třída
specifická věta:	uložení CT obrazu	→	SOP instance



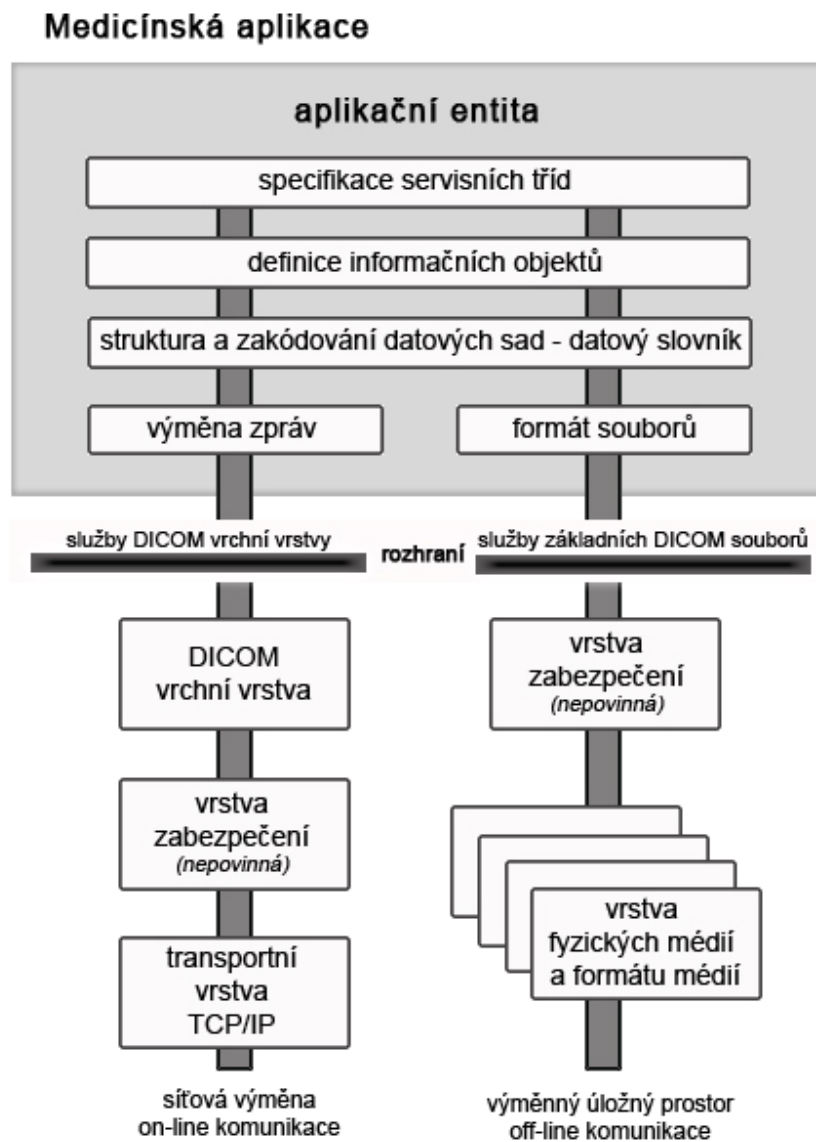
DICOM definuje řadu ukládacích SOP tříd: např. pro CT obrázky, MR obrázky atd. Do podoby SOP třídy pro ukládání CT obrázků je tedy zkombinována definice CT informačního objektu a třída ukládací služby. Při vytváření SOP instance jsou naplněny atributy informačních objektů a proměnné servisní třídy reálnými hodnotami (např. pacienta: jeho popis, snímací zařízení, snímky atd.) a instance dostane přiděleno UID (kap. 1.9).

Komunikace v DICOM standardu je reprezentována výměnou SOP instancí za pomoci SOP zpráv, což není nic jiného než komunikační verze SOP třídy. [12]

1.2.2 Hlavní komunikační model

Na Obr. 1.1 je ukázáno propojení dvou hlavních částí DICOM standardu, síťové části (on-line) a výměnného úložného prostoru (off-line). Aplikace mohou pro komunikaci využít dvou rozhraní:

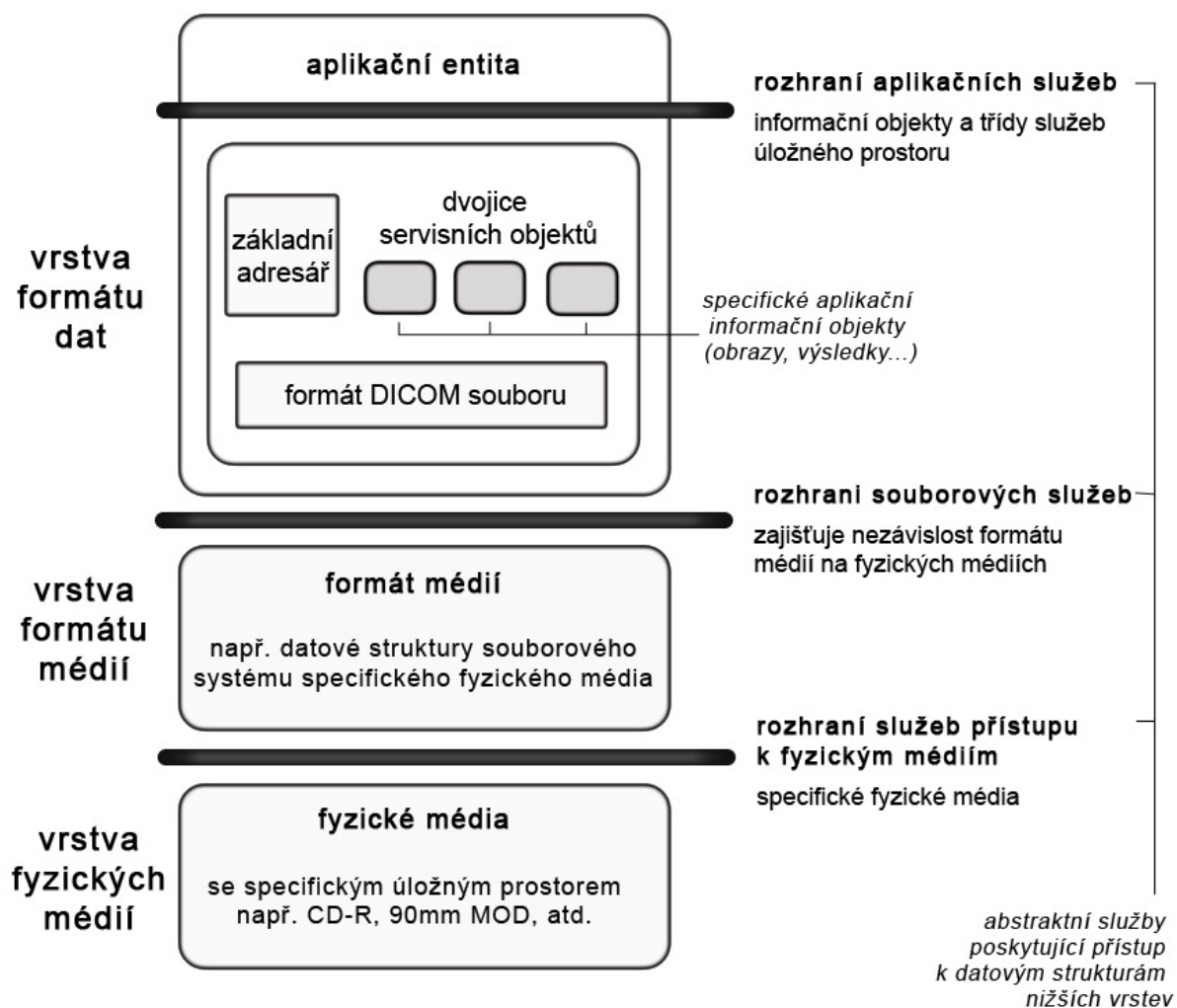
- Služby DICOM vrchní vrstvy, které poskytují nezávislost na specifickém fyzickém připojení do sítě a podporují komunikační protokoly, např. TCP/IP.
- Služby základních DICOM souborů, které poskytují přístup k úložnému prostoru nezávisle na specifickém formátu a struktuře souboru.



Obr. 1.1 Hlavní komunikační model DICOM

1.3 Model úložného prostoru

DICOM model úložného prostoru je charakterizovaný na Obr. 1.2 a rozšiřuje hlavní komunikační model Obr. 1.1. Zaměřuje se na hlediska přímo příbuzné s předáváním dat přes výměnný úložný prostor. Jedná se o datové struktury a přidružená pravidla užívaná v různých vrstvách k dosažení co nejlepší spolupráce. Služby určené v tomto modelu jsou jednoduchými rozhraními mezi funkčními vrstvami. V rozsahu tohoto modelu není specifikace rozhraní pro aplikační programy. DICOM model úložného prostoru zahrnuje tři vrstvy, které jsou popisovány v následujících podkapitolách.



Obr. 1.2 Model úložného prostoru

1.3.1 Vrstva fyzických médií (Physical Media Layer)

Jsou zde definované charakteristiky fyzické vrstvy, zahrnují činitele tvaru fyzických médií, rozsah, mechanické charakteristiky a záznamové vlastnosti. Tato vrstva také definuje seskupení zaznamenaných bitů.

Příkladem vrstvy fyzických médií v osobních počítačích je CD-R. Několik variant fyzických médií pro dané aplikace je v části PS 3.12 [9], v PS 3.11 [8] je definována řada aplikačních profilů,

kteře vyberou specifické fyzické médium v závislosti na požadavcích specifických lékařských zobrazovacích aplikací.

1.3.2 Vrstva formátu médií (Media Format Layer)

Ve vrstvě formátu médií jsou bitové toky fyzických médií organizované do specifických struktur. Datové struktury souboru a přidružené směrodatné struktury jsou definované k účinnému přístupu a správě prostoru fyzického prostředí.

Tato vrstva bývá často specifická pro prostředí každého operačního systému. Příkladem takové definice vrstvy formátu médií pro CD-R je datová struktura souborového systému osobního počítače náležící specifickému operačnímu systému. Několik voleb formátu médií je v části PS 3.12 [9].

Formáty médií podporované DICOM standardem jsou vybrány, aby podporovaly minimální požadavky specifikované službami DICOM souborů (kap. 1.5). Omezený přístup do souboru skřz tuto službu zajišťuje, že vrstva formátu dat je nezávislá na výběru fyzického média a jeho formátu.

1.3.3 Vrstva formátu dat (Data Format Layer)

DICOM vrstva formátu dat zahrnuje prvky specifikované:

- SOP třídou pro DICOM úložný prostor a příslušný IOD,
- formátem DICOM souboru,
- zabezpečeným formátem DICOM souboru,
- SOP třídou pro adresář DICOM úložného prostoru,
- aplikačním profilem pro DICOM úložný prostor,
- zabezpečeným profilem pro DICOM úložný prostor.

DICOM SOP třídy

DICOM SOP třídy a příslušné IOD jsou užívané pro zprostředkování specifických lékařských obrazových informací ve vrstvě formátu dat. SOP třídy a IOD užívané pro úložný prostor vyplývají z rámcového ustanovení v PS 3.3 [2] a PS 3.4 [3]. Příklady takových IOD jsou obrazové modality, pacientovy informace, výsledky, atd.

Použití DICOM IOD spolu se službami úložného prostoru tvoří řadu SOP tříd úložného prostoru nebo obecné SOP třídy. Služby úložného prostoru (např. čtení, zápis, vymazání, atd.) budou vykonány přes služby DICOM souboru. Koncept SOP tříd v souvislosti s úložným prostorem je rovnocenný ke konceptu SOP tříd pro související síťové operace (DIMSE Operations). V kontextu úložného prostoru mohou být užívané normalizované i kompozitní IOD a SOP třídy. Řada SOP tříd, které mohou být používány pro úložné prostory je definována v PS 3.4 [3], IOD na kterých jsou založeny jsou v přílohách DICOM PS 3.3 [2].

Koncept DICOM formátu souboru

Zapouzdření DICOM datové sady v souboru (kap. 1.4.2) definuje formát DICOM souboru schopný pojmout každou DICOM datovou sadu. Soubory jsou identifikované souborovým ID, ale z těchto ID (ani z jejich struktury) není odvozená žádná sémantika souboru. Lékařské zobrazovací aplikace ve funkci tvůrce DICOM souboru mohou užívat sémantickou informaci k tomu, aby generovaly ID souboru, ale zařízení (programy) které čtou DICOM soubory by se neměli spoléhat na zjevný sémantický obsah jejich ID.

Zapouzdření datové sady je založené na službách DICOM souboru (kap. 1.5). Specifický formát médií může nabízet větší souborové služby než jsou specifikované v službách DICOM souboru, ale jejich užívání je za rozsahem DICOM standardu. Rozšířené služby, které ohrožují součinnost, se nesmí používat (např. souborové ID delší než specifikované v službách DICOM souboru).

Zapouzdření DICOM souboru v zabezpečený DICOM soubor je specifikováno v podkapitole 1.4.4. Tyto pravidla definují mechanismus pro vytvoření zabezpečeného DICOM souboru zapouzdřením nechráněného DICOM souboru, i přes dané zesložení je to užitečné zatížení uvnitř bezpečné obálky.

DICOM informační lékařský adresář (Medical Information Directory)

K DICOM obrazům a obrazům souvisejících SOP tříd (např. výsledky, pacienti) mohou být navíc užívány další SOP třídy přizpůsobené úložnému prostoru pro poskytnutí referencí (nebo adresářů) založených na lékařské informaci, usnadní se tak přístup ke klinické obrazové informaci. Takovou SOP třídou je SOP třída pro adresář DICOM úložného prostoru, definovaná v PS 3.4 [3]. Příklady těchto SOP tříd jsou zprostředkované v souboru se souborovým ID z DICOMDIR (kap. 1.5.5).

1.3.4 DICOM aplikační profily úložného prostoru (Media Storage Application Profiles)

Aplikační profil úložného prostoru definuje výběr voleb v různých vrstvách DICOM modelu úložného prostoru, vhodné volby pro specifické potřeby nebo kontext, ve kterém se předpokládá že bude realizované prostředí výměny. Takové volby jsou formálně stanovené jako aplikační profil úložného prostoru, aby zajistily vzájemnou slučitelnost mezi implementacemi přizpůsobujícími se k stejnému aplikačnímu profilu úložného prostoru. To usnadňují úrovně shody (Conformance Statements), které dovolí uživateli určit vzájemnou slučitelnost různých implementací.

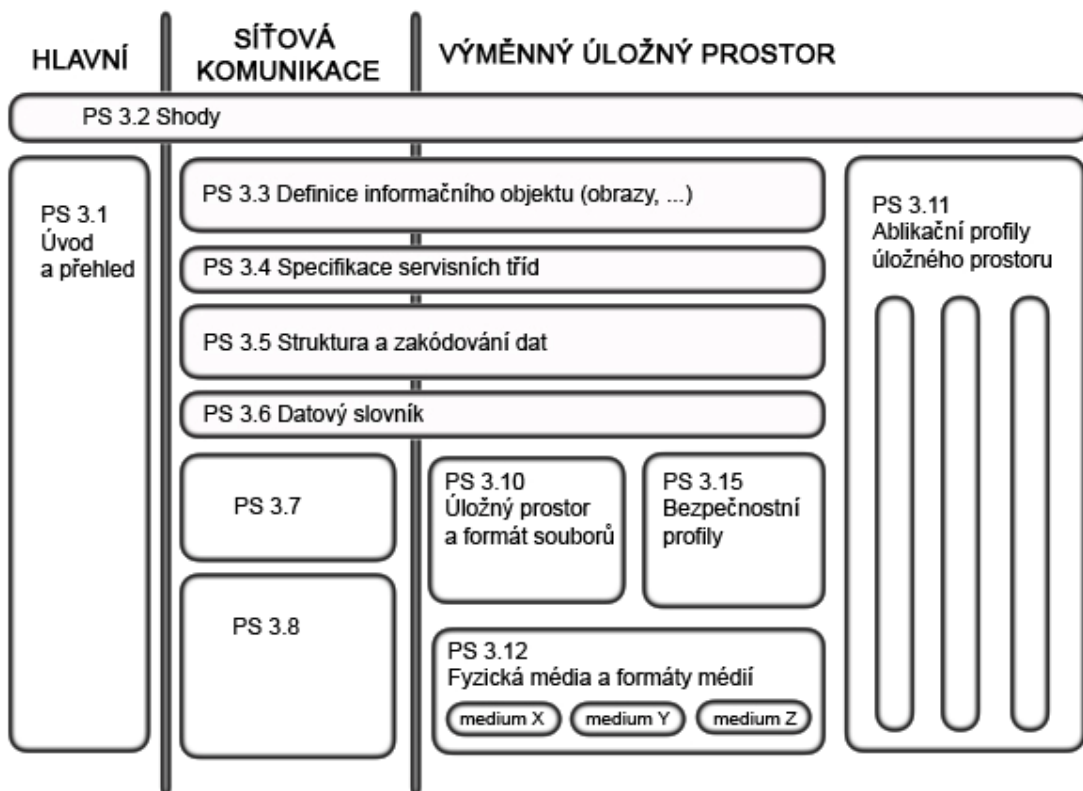
Aplikační profil úložného prostoru zahrnuje:

- Popis potřeby adresace aplikačním profilem (např., kardiak - trpící chorobou srdce).
- Ve vrstvě formátu dat výběr počtu specifický IOD a přidružených SOP tříd. Pro standardní DICOM SOP třídu, je to definováno v PS 3.4 [3]. Pro každou SOP třídu by mělo být uvedeno, jestliže je jeho podpora nutná nebo nepovinná v souvislosti s tímto profilem.
- Výběr specifických definic formátu médií. To je uděláno vztahem k PS 3.12 [9], který specifikuje vybrané fyzické médium, specifický přidružený formát média a mapování tohoto formátu média (systému souborů), které se stará o údržbu služeb DICOM souboru.
- Výběr vhodné přenosové syntaxe (kap. 1.10).
- Výběr specifického bezpečnostního profilu, tj. vztahem k PS 3.15 [10], který specifikuje kryptografické algoritmy používané při zapouzdření zabezpečených DICOM souborů.
- Další volby usnadnění vzájemné slučitelnosti jako specifického ohraničení (např. maxima velikosti souboru, bude-li to nezbytné).

Kompletní definice a struktura aplikačního profilu úložného prostoru je specifikována v PS 3.11 [8], kde je i řada standardních aplikačních profilů odpovídajících rozmanitým potřebám.

1.3.5 Úložný prostor a DICOM standardní struktura

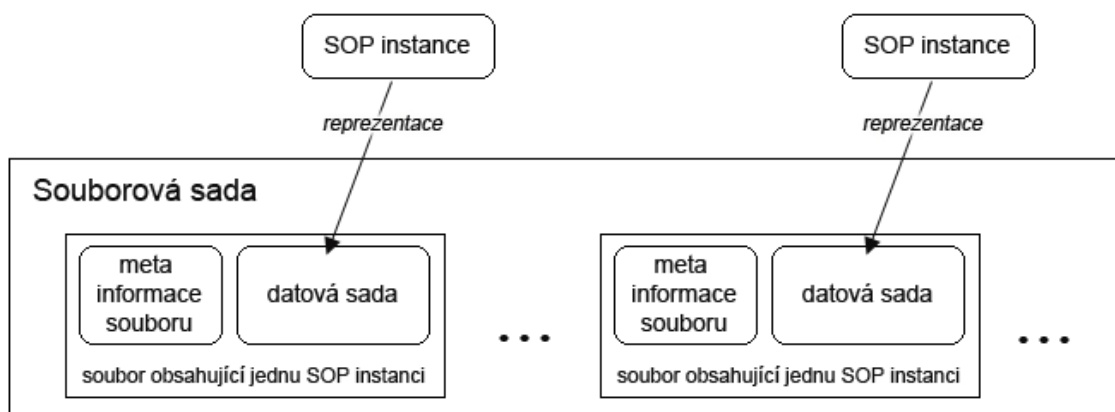
Obr. 1.3 poskytuje celkový pohled na vztahy mezi funkčními oblastmi určenými modelem úložného prostoru (Obr. 1.2) a různých částí DICOM standardu souvisejících s úložným prostorem. Řada částí DICOM standardu jsou společné pro síťovou komunikaci a výměnné média.



Obr. 1.3 Úložný prostor a DICOM části

1.4 Formát DICOM souboru

Formát souboru poskytuje prostředky k tomu, aby zapouzdřil datovou sadu (kap. 1.7) reprezentující SOP Instanci. Jak je ukázáno v Obr. 1.4, tok bajtů datové sady je umístěn do souboru po meta informacích. Každý soubor obsahuje jednotlivou SOP instanci.



Obr. 1.4 Formát souborů

1.4.1 Meta informace DICOM souboru

Meta informace souboru zahrnují identifikaci informací o zapouzdřené datové sadě (kap. 1.7.5). Tato hlavička se sestává z 128 bajtů preamble, 4 bajtů prefix, následují meta prvky souboru. Tato hlavička je přítomná v každém DICOM souboru.

Preamble je definovaný pro použití aplikačním profilem (viz. kap. 1.3.4) nebo jinou specifickou implementací. Nevyžaduje pevnou velikost ani strukturování jako DICOM datový prvek s tagem a délkou. Cílem toho je usnadnění přístupu k obrázkům a dalším datům v DICOM souboru, tedy kompatibilita s běžně užívanými souborovými formáty v počítačích. Pokud preamble není použit aplikačním profilem nebo specifickou implementací, všech 128 bajtů je nastaveno na 00H. Může ale např. obsahovat informace určené multimediálním aplikacím pro náhodný přístup k uloženým obrázkům.

Čtyři bajty DICOM Prefix budou obsahovat znakový řetězec „DICM“ zakódovaný jako znaky velkých písmen podle ISO 8859 G0 znakové sady (kap. 1.5.4). Tento čtyř bajtový prefix je uspořádaný jako DICOM datový prvek s tagem a délkou.

S výjimkou 128 bajtů preamble a 4 bajtů prefix budou meta informace souboru zakódovány použitím explicitní VR malého endianu přenosové syntaxe (UID=1.2.840.10008.1.2.1, kap. 1.10) jak definuje DICOM PS 3.5 [4]. Data každého prvku meta informací budou v případě potřeby vyplněna, aby dosáhla sudé délky odpovídající datové reprezentaci (Value Representation, zkráceně VR, kap. 1.6.2). Pro kompatibilitu s budoucími verzemi tohoto standardu, mohou být všechny tagy (0002,xxxx) ignorovány. Datové prvky se skupinou 0002 nebudou použité v datové sadě jinde, než uvnitř meta informací souboru.

1.4.2 Zapouzdření datové sady

Každý soubor bude obsahovat jednotlivou datovou sadu (kap. 1.7) představovanou jednotlivou SOP instancí (odvozené z SOP třídy a korespondující s IOD). Soubor přitom ale může obsahovat více 2D obrazových snímků, ve specifickém IOD může být definováno zahrnutí vícenásobných snímků.

Pro zakódování datové sady užívaná přenosová syntaxe (viz. kap. 1.10) bude rozpoznatelná podle UID přenosové syntaxe náležící meta informacím souboru.

DICOM datová sada nezahrnuje celkovou délku souboru. Jeho konec indikuje poskytnutá DICOM souborová služba (kap. 1.5), je to jediná indikace konce datové sady. Poslední prvek datové sady může být prvek (FFFC,FFFC), jestli je požadováno po zápisu souboru vyplnění datové sady (na sudé bajty). Data tohoto datového prvku (FFFC,FFFC) s koncovým vyplněním jsou neplatná a budou ignorována všemi DICOM implementacemi pro čtení této datové sady. Při čtení nebo aktualizaci souborové sady bude možno zpracovat tuto datovou sadu s koncovým vyplněním (FFFC,FFFC) definovanou buď meta informacemi nebo se musí jednat o datovou sadu vloženou v sekvenci.

1.4.3 Podpora práce se soubory

Formát DICOM souboru nezahrnuje informace o práci se soubory, aby se vyhnulo duplikaci s funkcemi příbuznými s vrstvou formátu médií. Následující informace, jsou-li nutné pro daný DICOM aplikační profil, by měly být nabízeny vrstvou formátu médií:

- identifikace vlastníka obsahu souboru,
- statistiky přístupu do souboru (např. datum a čas vytvoření),
- řízení přístupu k aplikačnímu souboru,
- řízení přístupu k fyzickému mediu (např. ochrana proti zápisu).

1.4.4 Zabezpečený formát DICOM souboru

Zabezpečený DICOM soubor bude obsahovat jednotlivé DICOM soubory zapouzdřené s Cryptographic Message Syntax jak je definováno v RFC 2630. V závislosti na kryptografických

algoritmech užívaných pro zapouzdření, zabezpečený DICOM soubor může poskytovat jednu nebo více následujících bezpečnostních vlastností:

- utajení dat (Data Confidentiality) – prostřednictvím šifrování,
- autentizace původu dat (Data Origin Authentication) – prostřednictvím certifikátů a digitálních podpisů,
- neporušenost údajů (Data Integrity) – prostřednictvím digitálních podpisů.

Navíc, zabezpečený DICOM soubor nabízí možnosti sdělovat šifrovací klíče a certifikáty k určeným příjemcům přes transportní klíč, klíčovou dohodu nebo souměrné klíče – klíčová šifrovací schémata.

1.5 Služby DICOM souboru (DICOM File Service)

Služby DICOM souboru specifikují abstraktní pohled na soubory z hlediska uživatele služeb ve vrstvě formátu dat (viz. kap. 1.3.3). Omezující přístup k obsahu souborů aplikačními entitami skrz takové služby DICOM souboru zajišťuje nezávislost vrstvy formátu dat fungující na specifickém formátu médií a výběru fyzického média.

Tato definice služeb DICOM souborů je výtah ve smyslu toho, že se jedná jen o specifikaci rozhraní. Jeho zaměření je omezené na hlediska přímo související s přístupem k datovým strukturám vrstvy formátu médií (ne specifikace svých datových struktur). Třebaže služby DICOM souboru mohou být popisované přes řadu abstraktních základních tvarů služeb jako je čtení, zápis, mazání, atd., nejsou zamýšleny jako definice rozhraní pro programování aplikací (Application Programming Interface, zkráceně API).

Služby DICOM souboru stanovené pro úložný prostor nabízí základní servis, dost jednoduchý aby byl podporován širokým okruhem obvykle dostupných formátů médií (nebo systémů souborů), ale dost bohatý aby poskytoval klíčové funkce k efektivnímu řízení souborů a přístupu k jejich obsahu. Následující podkapitoly specifikují minimální povinné požadavky, které budou uspokojeny každým fyzickým médiem a přidruženým formátem médií k tomu, aby vyhověly modelu DICOM úložného prostoru.

1.5.1 Souborová sada (File set)

Služby DICOM souboru nabízí schopnosti vytvořit a zpřístupnit jeden nebo více souborů v souborové sadě. Souborová sada je kolekce souborů, které uvnitř sdílejí společně pojmenovaný prostor jehož souborové ID (kap. 1.5.2) je unikátní. Žádná sémantika není spojena s pořadím souborů uvnitř souborové sady.

Příklad takového vztahu je mapování souborovým ID pojmenovaného prostoru k svazku formátu média v osobním počítači nebo k segmentu v systému souborů na výměnném médiu. Dalším příkladem je mapování souborovým ID pojmenovaného prostoru do adresářů a jeho stromových podadresářů. V tomto případě to může nabízet možnost podporování vícenásobné souborové sady (jedné na adresář) na stejném fyzickém médiu. Každá souborová sada má svůj vlastní DICOMDIR soubor. Pro zajištění vzájemné slučitelnosti, PS 3.12 [9] určuje tyto specifické mapující pravidla mezi adresáři a souborovým ID pojmenovaného prostoru souborové sady (včetně pravidel jak jednoznačně lokalizovat DICOMDIR soubor).

Jednotlivý soubor se souborovým ID DICOMDIR je zahrnutý v každé souborové sadě. Každá souborová sada je unikátně identifikovaná UID souborové sady (kap. 1.9), zaneseným do záznamů podle UID registračních předpisů stanovených v PS 3.5 [4]. Když jsou soubory přidány nebo odstraněny ze souborové sady, UID souborové sady se nezmění.

Souborová sada také může být identifikovaná ID souborové sady, který poskytuje jednoduchou (ale možná ne globálně unikátní) lidsky čitelnou referenci. ID souborové sady je řetězec od nuly do šestnácti znaků z podmnožiny G0 znakové sady ISO 8859 (kap. 1.5.4). ID souborové sady může být přidružený nebo mapovaný k vhodnému identifikátoru vrstvy formátu médií.

1.5.2 Souborové ID

Soubory jsou identifikované ID souboru, který je unikátní v souvislosti se souborovou sadou. ID souboru je seřazená sekvence souborových ID součástí, může obsahovat jednu až osm. Každá součást je řetězec jednoho až osmi znaků z podmnožiny G0 znakové sady ISO 8859 (kap. 1.5.4).

Taková struktura pro ID souboru (sekvence součástí) dovoluje službě DICOM souboru organizovat výběr souboru v hierarchickém módu. Nejsou definované žádné konvence DICOM standardu pro použití struktury souborových ID součástí a jejich obsahu (s výjimkou rezervovaného souborového ID DICOMDIR, kap. 1.5.5). Mimoto, sémantika nebude zprostředkována strukturou a obsahem takových ID souboru. Z toho vyplývá, když ID souboru bude přiřazeno každému souboru v souborové sadě, vytvářející DICOM aplikační entita si bude moct vybrat strukturu ID souboru, jakou si bude přát. Po AE, která čte existující soubory nebo vytváří nové soubory nebude požadované, aby věděla nějakou sémantiku originálního tvůrce (možná přidruženého s takovou strukturou).

1.5.3 Funkce a služby správy souboru (File Management Roles and Services)

Když se DICOM aplikační entity účastní výměny informací prostřednictvím úložného prostoru, vykonávají skrz služby DICOM souboru řadu služeb úložného prostoru: M-WRITE, M-READ, M-DELETE, M-INQUIRE FILE-SET, M-INQUIRE FILE.

DICOM aplikační entita se může ujmout jedné nebo více z následujících tří funkcí:

- Vytvoření souborové sady (File-set Creator, zkráceně FSC) prostřednictvím M-WRITE.
- Čtení souborové sady (File-set Reader, zkráceně FSR) prostřednictvím operace M-READ.
- Aktualizování souborové sady (File-set Updater, zkráceně FSU) prostřednictvím operací M-READ, M-WRITE, a M-DELETE.

Použití konceptu funkcí v úrovních shody (Conformance Statements) budou mít za následek přesnější vyjádření schopností implementací podporující DICOM úložný prostor. Souhlasné implementace budou podporovat jednu z kombinací uvedených funkcí (např. funkci FSC a FSR).

1.5.4 Znaková sada

ID souboru a ID souborové sady budou znakové řetězce zhotovené ze znaků z podmnožiny G0 souboru instrukcí ISO 8859. Tuto podmnožinu tvoří následující znaky:

- A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z (velká písmena)
- 1, 2, 3, 4, 5, 6, 7, 8, 9, 0 a _ (podtržítko)

Toto je znaková sada definovaná i pro Control Strings (VR CS - viz. kap. 1.6.2) až na to, že SPACE (znak mezery) zde není zahrnutý.

1.5.5 Rezervovaný ID souboru DICOMDIR (Reserved DICOMDIR File ID)

Jednotlivý soubor s ID souboru DICOMDIR existuje jako člen každé souborové sady. Toto ID souboru je zhotoveno z jednotlivých součástí (viz. kap. 1.5.2, struktura souborového ID). To obsahuje adresář DICOM úložného prostoru (viz. PS 3.3 [2] pro detailní specifikaci Basic Director IOD), který zahrnuje hlavní informaci o celé souborové sadě. Tato všeobecná informace je vždy přítomná, ale volitelně směrodatný obsah může být zanechaný prázdný v prostředích, kde by to nebylo potřebné. Příklad obsahu DICOMDIR souboru je v příloze A PS 3.10 [7].

DICOMDIR soubor užívá explicitní VR malého endianu přenosové syntaxe (UID = 1.2.840.10008.1.2.1, kap. 1.10) k zakódování SOP třídy adresáře DICOM úložného prostoru. Především:

- UID SOP třídy v meta informacích souboru (hlavička DICOMDIR souboru) bude mít hodnoty stanovené v PS 3.4 [3] pro SOP třídu adresáře DICOM úložného prostoru.
- UID SOP instance v meta informacích souboru (hlavička DICOMDIR souboru) bude obsahovat hodnotu UID souborové sady. UID souborové sady je přiřazený aplikační entitou, kterou vytvořila souborová sada (FSC funkce, viz. kap. 1.5.3) s žádným nebo více DICOM soubory.

Můžeme si představit, že tento postup odráží souborovou sadu jako abstrakci „kontejneru“, uvnitř kterého mohou být soubory vytvořeny nebo čteny. UID souborové sady souvisí s „kontejnerem“, ne jeho obsahem. Služby DICOM souboru v souborové sadě předpokládají, že budou mapovány k podpoře vlastností vybraných formátů média (např. velikost nebo oddíl).

1.6 Kódování dat

Sady dat jsou specifikované atributy IOD skutečných objektů, obsah a sémantika atributů je upřesněna v IOD (viz. PS.3.3 [2]). Struktura datových sad je v následující kapitole 1.7.

1.6.1 Podpora znakové sady pro grafické znaky

Hodnoty které jsou textové nebo znakové řetězce mohou být složeny z grafických a řídicích znaků. Grafická znaková sada, nezávislá na svém zakódování, se nazývá seznam znaků. V závislosti na jazyce, ve kterém se bude používat aplikační entita, se mění seznamy znaků používané v DICOM Standardu. Podporované jsou tyto: ISO 8859, JIS X 0201-1976, JIS X 0208-1990, JIS X 0212-1990, KS X 1001 (= ISO-IR 149), TIS 620- 2533 (1990), ISO 10646-1, 10646-2, GB 18030.

Standardní sada instrukcí pro znakové řetězce v DICOM je základní G0 sada International Reference verze ISO 646:1990 (ISO-IR 6). Viz. příloha E PS 3.5 [4], tabulka DICOM standardního souboru instrukcí a jeho zakódování. Tato základní G0 sada je identická s běžnou znakovou sadou ISO 8859 (viz. kap. 1.5.4).

DICOM aplikační entity mohou rozšiřovat nebo vyměňovat standardní soubor instrukcí, tuto informaci zprostředkovávají v atributu specifické znakové sady (0008,0005).

Standardní seznam znaků může být prodloužený nebo nahrazený. Pokud ano, příslušná „Specifická znaková sada“ bude definovaná jako atribut SOP všeobecného modulu (0008,0005) (viz. PS 3.3 [2]), použití bude uvedené v úrovních shody (Conformance Statement).

1.6.2 Datová reprezentace (Value Representation, zkráceně VR)

Datová reprezentace prvku dat popisuje typ dat a formát hodnot datových prvků. V PS 3.6 [5] je seznam VR každého datového prvku a náležitého tagu.

Hodnoty s VR tvoří znakové řetězce, kromě VR UI, kdy může být vyplněn znaky mezery (20H, v standardním seznamu znaků), aby dosáhl řetězec stanovené délky. Hodnoty s VR UI a VR OB budou doplněny jediným koncovým NULL (00H) znakem nezbytným k tomu, aby řetězec dosáhl stanovené délky. Všechny nové VR definované v budoucích verzích DICOM budou stejné struktury datového prvku jak je definováno v kapitole 1.7.1. Jednotlivá hodnota, včetně výplně, nebude přesahovat délku hodnoty, kromě případu poslední hodnoty vícehodnotového pole jak je specifikováno v následující podkapitole 1.6.3.

Tab. 1.2 Výňatek z datových reprezentací

Zkratka	Jméno	Zkratka	Jméno	Zkratka	Jméno
AE	Application Entity	IS	Integer String	SQ	Sequence of Items
AS	Age String	LO	Long String	SS	Signed Short
AT	Attribute Tag	LT	Long Text	TM	Time
CS	Code String	OB	Other Byte String	UI	Unique Identifier
DA	Date	OF	Other Float Sting	UL	Unsigned Long
DS	Decimal String	OW	Other Word String	UN	Unknown
DT	Date Time	PN	Person Name	US	Unsigned Short
FL	Floating Point Single	SH	Short String	UT	Unlimited Text
FD	Floating Point Double	SL	Signed Long		

Úplný seznam s popisem jednotlivých datových prvků je v PS 3.5 [4] strana 24-31.

1.6.3 Datová vícenásobnost (Value multiplicity, zkráceně VM) a ohraničení

Datová vícenásobnost datového prvku specifikuje počet hodnot, které mohou být zakódované v poli Data datového prvku. VM každého datového prvku je specifikovaná explicitně v PS 3.6 [5]. Jestli počet hodnot zakódovaných v prvku může být proměnný, bude to reprezentováno dvěma čísly oddělenými pomlčkou, například „1-10“ znamená, že rozsah dat jednoho prvku může být 1 až 10.

Když bude datový prvek obsahovat vícenásobná data, tak budou omezené tímto způsobem:

- pro znakové řetězce, znak 5CH (obrácené lomítko „\“ v souboru instrukcí ISO IR- 6) bude užívaný jako oddělovač mezi znakovými řetězci dat, ať budou pevné nebo proměnné délky,
- vícenásobné binární data pevné délky budou série spojovaných dat bez jakéhokoliv oddělovače.

Každý řetězec hodnot ve vícenásobném datovém znakovém řetězci může být sudé nebo liché délky, ale délka celého pole Data (včetně „\“oddělovače) bude sudé délky. Jestli je požadovaná výplň, aby pole Data bylo sudé délky, jednotlivý vyplňovací znak bude aplikovaný na konec pole Data (do posledních dat), v některém případě délka posledních dat může převyšovat délku dat o 1.

Jen poslední UID data ve vícenásobném datovém prvku dat s VR UI bude vyplněna jedním koncovým NULL (00H) znakem, který je nezbytný k tomu, aby zajistil, že celé pole dat (včetně „\“oddělovače) je stejné délky.

Prvky dat s VR SQ, OF, OW, nebo UI budou mít vždy datovou vícenásobnost v 1 (pro přehled datových reprezentací viz. PS 3.5 [4] strana 24-31).

1.7 Datová sada (Data Set)

Datová sada představuje instanci skutečného informačního objektu (Real World Information Object). Je zkonstruovaná z datových prvků a ty obsahují zakódované hodnoty atributů tohoto objektu. Obsah a sémantika těchto atributů jsou specifikovaná v IOD v PS 3.3 [2]. Tato kapitola se zaměřuje na konstrukci, charakteristické rysy, a zakódování datové sady a jejich prvků dat.

1.7.1 Prvky dat

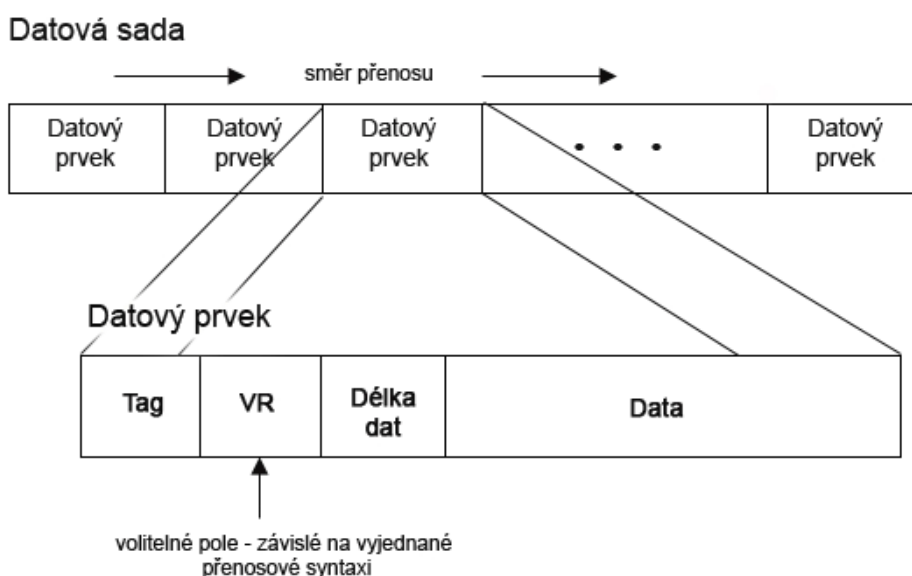
Datový prvek je jedinečně identifikovaný svým tagem. V datové sadě se prvky dat řadí podle zvyšujících čísel tagů. Unikátní číslo tagu datového prvku se bude vyskytovat nanejvýš jednou v datové sadě, ale ve vložených datových sadách se mohou čísla tagů opakovat.

Rozlišujeme dva definované druhy datových prvků:

- Standardní datové prvky mají sudé skupinové číslo s výjimkou čísel (0000,eeee), (0002,eeee), (0004,eeee) nebo (0006,eeee) které do této skupiny nepatří (jsou rezervované pro DIMSE příkazy viz. PS 3.7 [6])
- Privátní datové prvky mají liché skupinové číslo s výjimkou čísel které do tohoto druhu nepatří: (0001,eeee), (0003,eeee), (0005,eeee), (0007,eeee) nebo (FFFF,eeee). Privátní prvky dat jsou více rozvedeny v kapitole 1.7.6.

Datový prvek bude jedné ze tří struktur. Dvě z těchto struktur obsahují datovou reprezentaci (kap. 1.6.2) prvku a jsou tedy označovány za explicitní VR. Liší se ve způsobu, jak jsou vyjádřené jejich délky. Další struktura nazývaná jako implicitní VR neobsahuje datovou reprezentaci vůbec. Všechny tři struktury obsahují pro jeden datový prvek 3 stejné pole: Tag prvku dat, Délku dat a Data, jak je vidět na Obr. 1.5.

Implicitní a explicitní VR datového prvku neexistují současně v datové sadě ani datové sadě vložené (kap. 1.7.5). Jakou datovou reprezentaci datová sada používá je i s dalšími charakteristickými rysy určeno vyjednanou přenosovou syntaxí (kap. 1.10 a příloha A PS 3.5 [4]).



Obr. 1.5 Uspořádání DICOM datové sady a datových prvků

Pole datového prvku

Datový prvek je tedy vytvořený z těchto polí:

Tag datového prvku: Uspořádaná dvojice 16-bitových kladných celých čísel reprezentující číslo skupiny následované číslem prvku.

Datová reprezentace (VR): Dvoubajtový znakový řetězec obsahující VR datového prvku. Pro daný tag datového prvku bude definovaný datovým slovníkem, jak je specifikováno v PS 3.6 [5]. Znaky VR budou kódované DICOM standardní znakovou sadou.

Délka dat bude vyjádřena jednou ze dvou možností:

- Definovaná délka: 16 nebo 32 bitová délka (závisí na VR zda je explicitní nebo implicitní) kladné celé číslo obsahující explicitní hodnotu délky dat, tedy (sudý) počet bajtů tvořící data.
- Nedefinovaná délka: 32 bitová délka dat se nastaví na nedefinovanou (FFFFFFFFH). Může být používána pro datové prvky s VR SQ a UN, pro OW nebo OB záleží na vyjednané přenosové syntaxi (kap. 1.10 a příloha A PS 3.5 [4]).

Data: Sudé množství bajtů obsahujících data datového prvku. Typ dat uložených na tomto poli je specifikovaný VR datového prvku. VR pro daný tag datového prvku může být stanovená použitím datového slovníku v PS 3.6 [5], nebo použitím pole VR, jestli jej obsahuje explicitně uvnitř datového prvku. VR standardních datových prvků souhlasí se specifikovanými v datovém slovníku. Datová vícenásobnost (VM) specifikuje kolik hodnot s tímto VR může být umístěných v poli dat. Pole Data s nedefinovanou délkou jsou omezené pomocí sekvence ohraničení položky a položky ohraničení datového prvku, které jsou dále popsány v kapitole 1.7.5.

Struktura datového prvku s explicitní VR

Při používání explicitních struktur VR, bude datový prvek zkonstruovaný ze čtyř za sebou jdoucích polí: Tag datového prvku, VR, Délka dat a Data. V závislosti na VR datového prvku bude datový prvek uspořádaný následovně: 16 bitů následujících dvou znakového VR pole rezervovaných pro použití pozdějšími verzemi DICOM standardu. Tyto rezervované bajty budou nastavené na 0000H a nebudou užívané nebo dekodované (Tab. 1.3). Délka dat je 32-bitové kladné celé číslo.

Pro všechny další VR je pole délka dat 16-bitové kladné celé číslo následující dvou znakového pole VR (

Tab. 1.4). Pole Délka dat se rovná délce pole Data.

Tab. 1.3 Datový prvek s explicitní VR

Tag		VR		Délka dat	Data
Číslo skupiny (16-bitový unsigned integer)	Číslo prvku (16-bitový unsigned integer)	VR (2 byte znakovéh o řetězce) z "OB", "OW", "OF", "SQ", "UT" nebo "UN"	Reservováno (2 byte) nastaveno na hodnotu 0000H	32-bitový unsigned integer	Sudé množství bajtů obsahujících hodnotu datového prvku zakódovaných podle VR a vyjednané přenosové syntaxe. Data jsou omezená Sekvencí ohraničení položky, jestli jsou nedefinované délky.
2 byte	2 byte	2 byte	2 byte	4 byte	velikost 'Délky dat' v bajtech, jsou-li data explicitní délky

Tab. 1.4 Datový prvek s jinou explicitní VR než je v Tab. 1.3

Tag		VR	Délka dat	Data
Číslo skupiny (16-bitový unsigned integer)	Číslo prvku (16-bitový unsigned integer)	VR (2 byte znakového řetězce)	(16-bitový unsigned integer)	Sudé množství bajtů obsahujících hodnotu datového prvku zakódovaných podle VR a vyjednané přenosové syntaxe.
2 byte	2 byte	2 byte	2 byte	velikost 'Délky dat' v bajtech

Struktura datového prvku s implicitní VR

Při používání implicitních VR struktur bude datový prvek zkonstruovaný ze tří za sebou jdoucích polí: Tag datového prvku, Délka dat a Data (viz. Tab. 1.5). Jestli pole Data má explicitní délku, pak pole Délka dat bude obsahovat hodnotu rovnou délce (v bajtech) pole Data. Neboli pole Data má nedefinovanou délku a sekvence vymezení položky označuje konec pole Data.

Tab. 1.5 Datový prvek s implicitní VR

Tag		Délka dat	Data
Číslo skupiny (16-bitový unsigned integer)	Číslo prvku (16-bitový unsigned integer)	32-bitový unsigned integer	Sudé množství bajtů obsahujících hodnotu datového prvku zakódovaných podle VR specifikované v PS 3.6 a vyjednané přenosové syntaxe. Data jsou omezená Sekvencí ohraničení položky, jestli jsou nedefinované délky.
2 byte	2 byte	4 byte	velikost 'Délky dat' v bajtech, jsou-li data nedefinované délky

1.7.2 Délka skupiny (Group Length)

Délka skupiny (gggg,0000) standardního datového prvku je vysloužilá. Viz. PS 3.5 2007.

Všechny implementace budou schopné rozebrat elementy délky skupiny, vyřadit je, ale nikoli je vložit. Implementace nebude požadovat přítomnost elementů délky skupiny.

Je doporučeno, aby elementy délky skupiny byly odstraněny během uložení nebo přenosu, aby se vyhnulo riziku neslučitelnosti u hodnot datového prvku a změn v přenosových syntaxích.

1.7.3 Řazení bajtů (Byte Ordering), velký a malý endian

Další komponenta kódování datové sady, která bude souhlasit s komunikací aplikačních entit je řazení bajtů.

Řazení bajtů podle **malého endianu** je definováno tímto způsobem:

- V binárním čísle sestávajícím se z vícenásobných bajtů (např. 32-bitová hodnota kladného celého čísla, skupinové číslo, číslo prvku, atd.), bude nejprve zakódovaný nejméně významný bajt, zbývající bajty budou zakódovány v pořadí zvyšujícího se významu.
- Ve znakovém řetězci sestávajícím se z vícenásobných 8-bitových jednotlivých bajtů kódu, budou znaky zakódovány v pořadí výskytu v řetězci (zleva doprava).

Řazení bajtů podle **velkého endianu** je definováno tímto způsobem:

- V binárním čísle sestávajícím se z vícenásobných bajtů, nejvýznamnější bajt bude zakódovaný nejprve, zbývající bajty budou zakódovány v pořadí stále menšího významu.
- Ve znakovém řetězci sestávajícím se z vícenásobných 8-bitových jednotlivých bajtů kódu, budou znaky zakódovány v pořadí výskytu v řetězci (zleva doprava).

Řazení bajtů je komponenta dohodnutá přenosovou syntaxí (kap. 1.10). Standardní DICOM přenosové syntaxe, které budou podporovány všemi AE, užívají kódování malého endianu a jsou specifikovány v příloze A.1 PS 3.5 [4]. V této příloze jsou specifikovány i alternativní přenosové syntaxe, z kterých některé užívají kódování velkého endianu.

V standardním případě kódování malým endianem, stroje interpretující sadu dat velkého endianu udělají „bajtový přesun“ („byte swapping“) před interpretováním nebo operováním na určitých datových prvcích. Zasažené datové prvky jsou všechny ty, které mají VR vícenásobné bajtové hodnoty a které nejsou znakové řetězce z 8-bitových jednotlivých bajtů kódů, proto nejsou ovlivněny řazením bajtů. VR které nejsou řetězce znaků a skládají se z vícenásobných bajtů jsou:

- 2-bajtové US, SS, OW a každá komponenta z AT,
- 4-bajtové OF, UL, SL, a FL,
- 8-bajtové FD.

Pro výše uvedené VR, jsou vícenásobné bajty v pořadí zvyšujícího se významu ve formátu malého endianu. Například, 8-bajtový datový prvek s VR FD, by mohl být napsaný šestnáctkově 68AF4B2CH, ale v kódování malého endianu by byl 2C4BAF68H.

1.7.4 Typ datového prvku

Jestli atribut, zakódovaný jako datový prvek, může nebo nemůže být požadovaný v datových sadách, závisí na tomto typu atributu datového prvku (Attribute's Data Element Type).

Typ datového prvku jako atributu IOD nebo atribut SOP třídy je užíváný pro specifikování zda je atribut povinný nebo volitelný. Typ datového prvku také signalizuje, jestli je atribut podmíněný (jen povinný za určitých podmínek). Typy datových prvků atributů kompozitních IOD jsou specifikované v PS 3.3 [2]. Typy datových prvků atributů normalizovaného IOD jsou specifikované jako atributy SOP tříd v PS 3.4 [3].

- **typ 1** – definuje povinné prvky, pole Data bude obsahovat platné data definované VR a VM,
- **typ 1C** – prvky obsaženy za určitých podmínek, jinak stejný jako typ 1,
- **typ 2** – definuje povinné prvky, dovoluje neznámé hodnoty (nulová délka dat, žádné data), jinak stejný jako typ1,
- **typ 2C** – prvky obsaženy za určitých podmínek, jinak stejný jako typ 2,
- **typ 3** – nepovinný, nepřítomnost není porušením protokolu jako u předešlých typů,
- **typ uvnitř sekvence** – pro vnořené datové sady.

1.7.5 Vnořené datové sady

VR identifikovaný SQ je užíváný pro datové prvky s hodnotou sestávající se ze sekvence nul nebo více položek, kde každá položka obsahuje sadu datových prvků. SQ poskytuje flexibilní kódové schéma, které může být používáné pro jednoduché struktury opakování sad datových prvků, nebo kódování více komplexních IOD často nazývaných složky. SQ datové prvky mohou také užívat rekurzi, aby obsahovaly víceúrovňové vnořené struktury.

Položky součástí SQ datového prvku jsou uspořádány jako množina, kde každá položka může být referencí jeho pořadové pozice. Každé položce je implicitně přiřazena pořadová pozice počínaje hodnotou 1 pro první položku v sekvenci a zvýšená o 1 s každou následující položkou.

Jednotlivé vnořené datové prvky vycházejí z definic v kapitole 1.7.1, jejich konstrukce je patrná z jednotlivých příkladů:

Tab. 1.6 Příklad datového prvku s implicitní VR SQ s 3 položkami explicitní délky

Tag datového prvku	Délka datového prvku	Data datového prvku								
(gggg, eeee) s VR SQ	0000F0 0H	První položka			Druhá položka			Třetí položka		
		Tag položky (FFFE, E000)	Délka položky 0000 04F8H	Data položky datová sada	Tag položky (FFFE, E000)	Délka položky 0000 04F8H	Data položky datová sada	Tag položky (FFFE, E000)	Délka položky 0000 04F8H	Data položky datová sada
4 byte	4 byte	4 byte	4 byte	04F8H byte	4 byte	4 byte	04F8H byte	4 byte	4 byte	04F8H byte

Tab. 1.7 Příklad datového prvku s explicitní VR SQ nedefinované délky, obsahující 2 položky explicitní délky

Tag datového prvku	Datová reprezentace		Délka datového prvku	Data datového prvku							
(gggg, eeee) s VR SQ	SQ	0000H Reservováno	FFFF FFFFH nedef. délky	První položka			Druhá položka			Sekvence vymezení položky	
				Tag položky (FFFE, E000)	Délka položky 98A5 2C68H	Data položky datová sada	Tag položky (FFFE, E000)	Délka položky B321 762CH	Data položky datová sada	Tag vmez. sekv. (FFFE, E0DD)	Délka položky 0000 0000H
4 byte	2 byte	2 byte	4 byte	4 byte	4 byte	98A5 2C68H byte	4 byte	4 byte	B321 762CH byte	4 bytes	4 bytes

1.7.6 Privátní datové prvky (Private data elements)

Implementace mohou vyžadovat komunikační informace, která nemohou být obsaženy v standardních datových prvcích. Proto pro takové informace budou využívány privátní datové prvky, které nemění sémantiku z IOD nebo SOP třídy.

Privátní datové prvky mají stejnou strukturu jako standardní datové prvky specifikované na začátku kapitoly 1.7.1. Skupinové číslo užívané v tagu privátních datových prvků bude liché číslo. Privátní datové prvky budou obsaženy v datové sadě ve vzestupně řazených číslech podle tagu datového prvku. Pro typy datových prvku 1, 1C, 2 nebo 2C nemohou být užívány privátní datové prvky namísto požadovaných standardních datových prvků.

Není vyloučeno, že různí realizátoři mohou definovat privátní prvky se stejným (lichým) skupinovým číslem. Aby se vyhnulo konfliktům, tak se k privátním prvkům přiřazují tagy privátních datových prvků podle specifických pravidel.

VR užívaná pro privátní datové prvky bude stejná jako VR specifikovaná pro standardní datové prvky v kapitole 1.6.2. Zakódování je přizpůsobeno podle požadavků pro VR a jsou v souladu s vyjednanou přenosovou syntaxí. Privátní datový prvek s VR SQ (privátní datová sekvence) může

zahrnovat položky se standardními ale i privátními datovými prvky. Standardní datové prvky užívané uvnitř privátní datové sekvence budou užívat VR definovaný v datovém slovníku PS 3.6 [5].

Pro standardní prodloužené SOP třídy nebudou atributy Pixel data, Waveform data a Overlay data obsažené uvnitř privátní sekvence položek, ani uvnitř standardní sekvence položek vložené přímo nebo nepřímo uvnitř privátní sekvence položek.

1.8 Zakódování datových prvků: Pixel, Overlay a Waveform data

Pixel datový prvek (7FE0,0010) a Overlay datový prvek (60xx,3000) jsou užívané pro výměnu zakódovaných grafických obrazových dat. Tyto prvky spolu s dodatečnými datovými prvky, specifikovanými atributy obrazové informační entity (definované v PS 3.3 [2]), jsou užívané pro popis cesty ve které jsou Pixel data a Overlay data zakódovaná a jak budou interpretována. V závislosti na vyjednané přenosové syntaxi (kap. 1.10 a příloha A PS 3.5) mohou být Pixel data komprimována.

Pixel datový prvek a Overlay datový prvek má VR OW nebo OB, v závislosti na vyjednané přenosové syntaxi. Jediný rozdíl mezi OW a OB je, že řetězec bajtů OB nebude ovlivňován řazením bajtů (viz. kap. 1.7.3).

1.8.1 Zakódování Pixel dat ze souvisejících datových prvků

Zakódovaná Pixel data mohou být přizpůsobené různým bitovým hloubkám. Následující tři datové prvky budou definovat pixel strukturu:

- alokované bity (Bits Allocated) (0028,0100),
- uložené bity (Bits Stored) (0028,0101),
- horní bit (High Bit) (0028,0102).

Každá pixel buňka bude obsahovat jednotlivou hodnotu pixel vzorku. Velikost pixel buňky bude specifikovaná alokovanými bity. Uložené bity definují celkový počet těchto alokovaných bitů, které budou užívané pro reprezentaci hodnoty pixel vzorku. Uložené bity nebudou nikdy větší než bity alokované. Horní bit specifikuje, jak má být uzpůsobeno řazení uložených bitů s ohledem na specifikaci alokovaných bitů. Bity neužívané pro hodnoty pixel vzorků mohou být použité pro plánované překrytí (overlay planes, PS 3.3 [2]).

Například, v Pixel datech s 16 bity (2 bajty) alokovanými, bude 12 bitů uložených a bit 15 specifikovaný jako horní. Jeden pixel vzorek je tedy zakódovaný v 16 bitovém slově s 4-mi LSB každého slova neobsahující pixel data, viz. příloha D PS 3.5 [4].

Omezení pro tyto 3 datové prvky definující pixel strukturu jsou specifikované v IOD v PS 3.3 [2], pole Data obsahující Pixel data může být na konci vyplněno na sudou bajtovou hodnotu.

V multi-rámcovém objektu který je přenesený v nativním formátu nejsou jednotlivé rámce vyplňované. Individuální rámce budou kaskádní a vyplňovací bity (bude-li to nezbytné) se aplikují na doplnění polí Data.

Pole bitů představované hodnotou pixel vzorku bude binární celé číslo dvojkového doplňku nebo kladné celé číslo, jak je specifikováno datovým prvkem reprezentace pixelu (0028,0103). Znaménkový bit bude horní bit v hodnotě pixel vzorku, ta jest celé číslo dvojkového doplňku. Minimální aktuální hodnotou pixel vzorku je obsažena v pixel datech, které jsou specifikované

nejmenší hodnotou obrazového pixelu (0028,0106) zatímco maximální hodnota je specifikovaná největší hodnotou obrazového pixelu (0028,0107).

1.8.2 Zakódování Overall dat ze souvisejících datových prvků

Zakódované Overlay Planes (vrstvy rovin) mají vždy bitovou hloubku 1 a jsou zakódovány odděleně od Pixel dat v Overlay datech (60xx,3000). Následující dva datové prvky budou definovat strukturu Overlay Plane (vrstvy roviny):

- Overlay alokované bity (Overlay Bits Allocated) (60xx,0100),
- Overlay pozice bitu (Overlay Bit Position) (60xx,0102).

Od roku 2004 (PS 3.5-2004) je pro Overlay zakódované v Overlay datovém prvku (60xx,3000) rovny Overlay alokované bity hodnotě 1 a Overlay pozice bitu je v 0.

Pro Overlay datové prvky (60xx,3000) je nejčastěji požadováno VR OW. V případech kde VR je explicitně zprostředkované (příloha A PS 3.5 [4]) může být pro Overlay data užíváno VR OB.

1.8.3 Nativní nebo zapouzdřený formát zakódování

Pixel data zprostředkovaná v pixel datovém prvku (7FE0,0010) mohou být v nativním formátu (nekomprimovaném) nebo v zapouzdřeném formátu (komprimovaném) definovaném vně DICOM standardu.

Jestli budou Pixel data v nativním formátu, je nejčastěji požadována VR OW. VR OB může být také užívána pro Pixel data, ale jen v případech, kde alokování bitů má hodnotu menší nebo rovno 8 s přenosovou syntaxí kde je VR explicitní (příloha A PS 3.5 [4]).

Nativní formát pixel buňky (Pixel Cell) je zakódovaný přímým zřetěžením bitů každé pixel buňky, nejnižší platný bit (LSB) každé pixel buňky je LSB kódovaného slova nebo bajtu. Počet bitů každé pixel buňky je definovaný alokovanými bity (0028,0100) dat datového prvku.

Jestli je užíváno VR OB poslané v zapouzdřeném formátu, pixel buňka je zakódována podle kódujícího procesu definovaného jedním z vyjednaných přenosových syntaxí (příloha A PS 3.5 [4]). Zapouzdřený tok pixelů kódovaných pixel data je rozdělený do jednoho nebo více fragmentů, které zprostředkují jejich explicitní délku. Sekvence fragmentů zapouzdřeného toku pixelů je ukončena oddělovačem, jak dovoluje podpora kódujících procesů, kde výsledná délka celého toku pixelů je známá do doby, než je zcela zakódována. Tento zapouzdřený formát podporuje oba Single-Frame i Multi-Frame obrazy (PS 3.3 [2]).

Zapouzdřený formát poskytuje podporu pro použití normalizovaných kompresních mechanismů, jako jsou např. JPEG, JPEG-LS, JPEG 2000, RLE, MPEG2 MP@ML a pro podporu přenosu obrazů přes síť JPIP (PS 3.5 [4], sekce 8.2 a 8.4).

1.9 Unikátní identifikátory (Unique Identifiers, zkráceně UID)

Unikátní identifikátory poskytují kompatibilitu k jedinečnému identifikování široké škály položek. Garantují jedinečnost napříč rozmanitými zeměmi, prodavači a vybavením. Různé třídy objektů, instance objektů a informační entity se mohou rozeznávat přes DICOM univerzální řeč bez ohledu na sémantický kontext.

UID schéma identifikace je založené na OSI identifikaci objektu (numerické formy) jak definuje ISO 8824 standard. Všechny jednoznačné identifikátory, užívané v souvislosti s DICOM standardem, jsou zaneseny do záznamů hodnoty, jak definuje ISO 9834-3 pro zajištění globální jedinečnosti.

Každý UID je složený ze dvou částí:

UID = <kořen>.<přípona> (UID = <org root>.<suffix>)

- <kořen> jedinečně identifikuje organizaci (např. výrobce, výzkumnou organizaci, NEMA, atd.) a je složený z několika číselných částí jak definuje ISO 8824,
- <přípona> je také složená z několika numerických částí a bude unikátní uvnitř rozsahu <kořene>, z toho vyplývá, že organizace identifikovaná v <kořenu> je odpovědná za garantování jedinečné <přípony>.

<kořen> „1.2.840.10008“ je rezervovaný pro DICOM definované položky (např. DICOM přenosová syntaxe) a není užívaný pro privátně definované položky (např. instance obrazu).

Ačkoli specifická implementace si může vybrat nějakou specifickou strukturu pro vygenerování UID, nemělo by se nikdy předpokládat, že UID nese nějakou sémantiku. Definice komponenty (pro příponu) je specifická implementace a může se změnit pokud je udržena jedinečnost. Syntaktická analýza UID může ohrozit schopnost součinnosti, vyvíjení implementací.

Příklad UID struktury je v příloze C PS 3.5 [4].

1.10 Přenosová syntaxe

Přenosová syntaxe je sada kódujících pravidel schopných jasně znázorňovat jednu nebo více abstraktních syntaxí. Zvláště dovoluje komunikujícím aplikačním entitám (AE) vyjednávat společné kódovací techniky (např. řazení bajtů, komprese, atd.). Přenosová syntaxe je atribut prezentačního kontextu (Presentation Context), který je vyjednaný v ustanoveních asociací mezi DICOM AE (specifikováno v PS 3.3 [2] a částečně v PS 3.7 [6]).

Výběr přenosové syntaxe platí pro kódovací pravidla jen pro datovou sadu DICOM zpráv. Všechny DICOM standardní a privátní přenosové syntaxe implicitně specifikují pevné zakódování pro sadu příkazů (Command Set) DICOM zpráv jak je specifikováno v PS 3.7 [6]. Standardní DICOM přenosová syntaxe je specifikována v příloze A PS 3.5 [4]. DICOM zobrazení pro jméno přenosové syntaxe je zobrazení užívané pro UID (viz. předchozí kapitola 1.9).

Organizace zodpovědná za definici a registraci DICOM přenosové syntaxe je NEMA. Ta garantuje jedinečnost pro všechny jména DICOM přenosové syntaxe. Privátně definovaná jména přenosové syntaxe mohou být také užívána, ale musí se shodovat s registračními postupy, přesto nejsou organizací NEMA evidovány.

DICOM definuje standardní přenosovou syntaxi, DICOM implicitní VR malého endianu přenosové syntaxe (UID = „1.2.840.10008.1.2“), která bude podporována každou shodnou DICOM implementací.

Dále definuje specifické přenosové syntaxe pro např. JPEG, RLE, JPEG-LS, JPEG 2000, MPEG2 MP@ML, JPIP (PS 3.5 [4], sekce 10.2 až 10.8).

2 Popis výsledného programu

Aplikace byla vytvářena v jazyce C, C++, který je mezi vývojáři značně rozšířený. S tím souvisí dobrá dostupnost překladačů, nápověd a vzorových ukázek kódu pro různé platformy. Vývojovým prostředím bylo Microsoft Visual Studio 2005 na platformě Windows verze XP. Jako rozšíření pro přehlednou práci s DICOM formátem byla použita open source knihovna Imebra verze 0.0.46 [14]. Vzhled aplikace byl vytvořen za pomoci grafické nástavby CLR.

2.1 Imebra

Jedná se o knihovnu, která je k dispozici zdarma pod licencí GNU AFFERO GENERAL PUBLIC LICENSE (verze 3, z 19. listopadu 2007). Jejím tvůrcem je Paolo Brandoli a Copyright vlastní společnost Puntoexe.

Knihovna byla navržena k tomu, aby sloužila jako handle (objekt sloužící jako ukazatel, reference) DICOM soborů, zahrnuje ale také třídy pracující s obrazovými soubory. Jejimi charakteristickými základními rysy jsou:

- multi-platformost,
- samostatnost (knihovna užívá jen funkce ze standardní C a template knihovny),
- podpora DICOM a JPEG obrazové dekomprese a komprese,
- konverze barevného prostoru,
- podpora pro vložené datové sady (sekvence) a transakce.

2.2 Přidání knihovny Imebra do projektu

Imebra není distribuovaná jako dynamická jednotka a pravděpodobně ani nikdy nebude. Je to kvůli jazyku C++ užívanému pro vytváření knihoven a kvůli různým kompilačním programům, které užívají různé konverze.

Kvůli tomu, musí být celé Imebra zdrojové soubory spojené s vytvářeným projektem. Pokud je užíván stejný C++ kompilační program k tomu, aby vytvářel aplikaci a Imebra knihovnu, bylo by možné vytvořit dynamickou jednotku, která by zahrnovala Imebra knihovnu.

Imebra hojně využívá stl knihovnu, pokud ji vývojové prostředí neposkytuje, je nutno ji instalovat např. z stlport (<http://www.stlport.org>). Dále je potřeba standardní C knihovna.

Pro práci s knihovnou ve vývojovém prostředí v desktopovém operačním systému Windows musí být v seznamu cest hledaných „include“ příkazem cesta k `window.h` souboru a definovaný symbol `_WIN32_WINNT` v C++ preprocesoru na hodnotu `0x0400` nebo větší (pro Windows CE nemusí být definován).

Aby knihovna pracovala, musí být projektu přidány všechny `cpp` zdrojové soubory v adresářích `puntoexe_library/imebra/src` a `puntoexe_library/base/src` obsažené v distribučním balíku. Pro všechny Imebra zdrojové soubory je nutné vypnout používání předkompilovaných hlaviček. Hlavičkové soubory v `puntoexe_library/imebra/include` a `puntoexe_library/base/include` nemusí být v projektu zahrnuty, ale žádná škoda se nestane, pokud budou přidány.

Kompilátor musí podporovat RTTI.

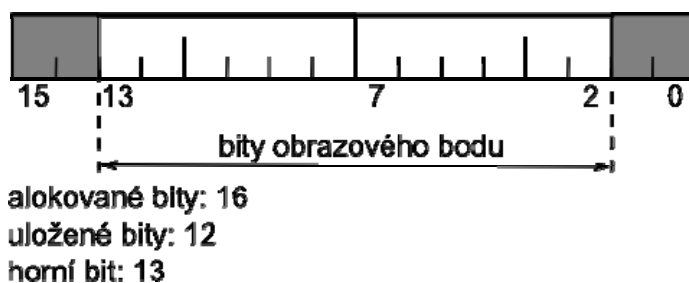
2.3 Uložení informace

Pro zjištění skutečného uspořádání bajtů na fyzickém médiu je důležitá přenosové syntaxe (viz. kap. 1.10) z nichž základní jsou:

- UID = 1.2.840.10008.1.2 - implicitní datová reprezentace malého endianu,
- UID = 1.2.840.10008.1.2.1 - explicitní datová reprezentace malého endianu,
- UID = 1.2.840.10008.1.2.2 - explicitní datová reprezentace velkého endianu,
- UID = 1.2.840.10008.1.2.4.XX - různé druhy JPEG kompresí,
- UID = 1.2.840.10008.1.2.4.94 (a 95) - určení JPIP,
- UID = 1.2.840.10008.1.2.4.100 - komprese MPEG2 Main Profile @ Main Level.

Podrobný přehled všech definovaných UID přenosových syntaxí je v PS 3.6 [5], příloha A.

Jak bylo popsáno v kapitole 1.8.1, přehled o bitech nesoucích obrazovou informaci lze získat z tagů: alokované bity, uložené bity, a horní bit.

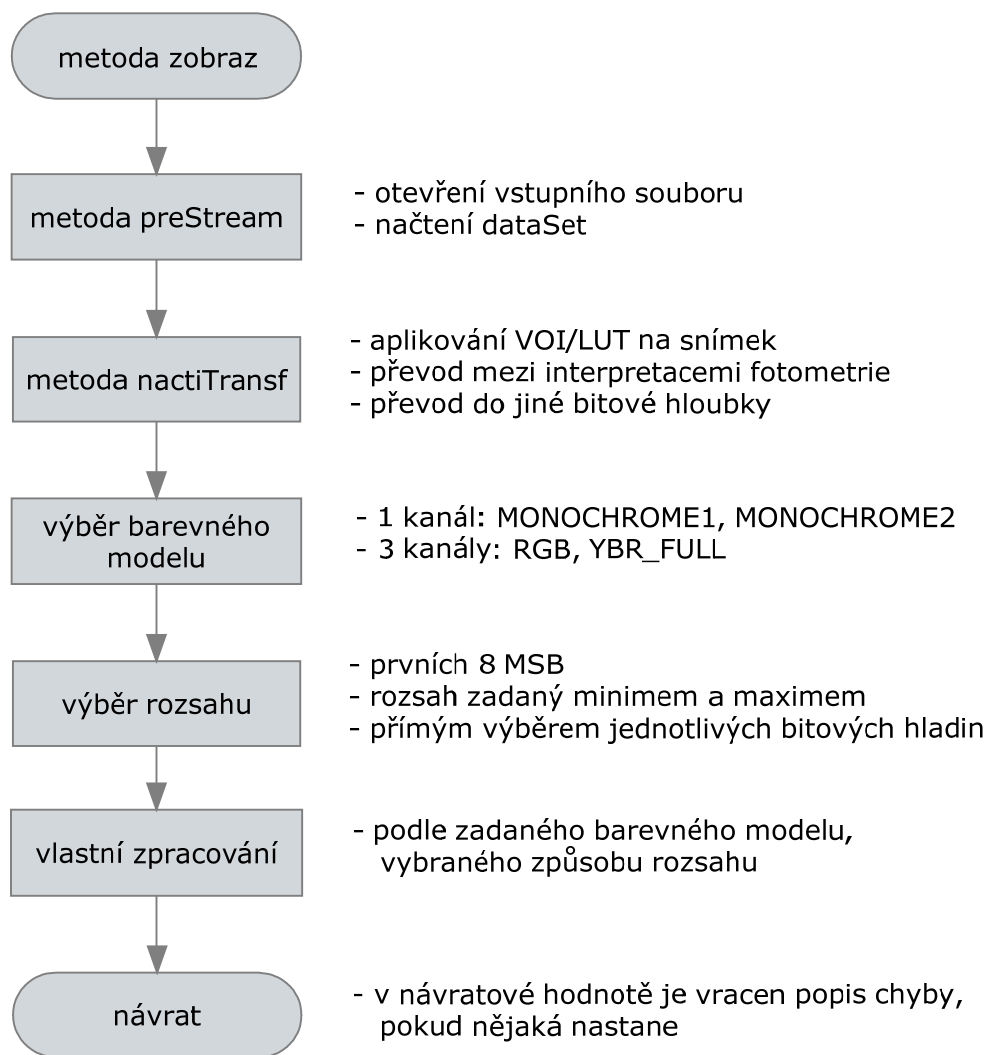


Obr. 2.1 Příklad uložení pixelu

2.4 Struktura programu

Jádrem programu je třída `MyClassImebra`. Nachází se v ní soukromé (`private`) a veřejné (`public`) pomocné ukazatele knihovny `Imebra`. Z veřejně přístupných částí jsou nejdůležitější:

- dva ukazatele:
 - `myDataSet`,
 - `myImage`,
- dvě metody:
 - `zobraz` – do bitmapy předané odkazem ukládá výslednou grafickou prezentaci uložených bitů podle nastavení aplikace,
 - `uloz` – ukládá soubory ve formátu `jpeg` nebo `dicom`.

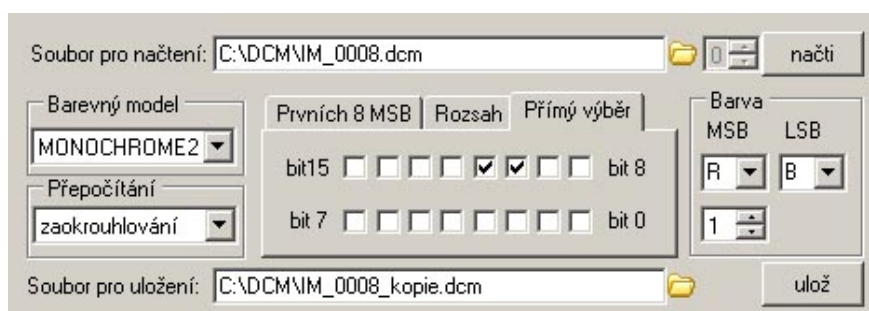


Obr. 2.2 Vývojový diagram metody zobraz

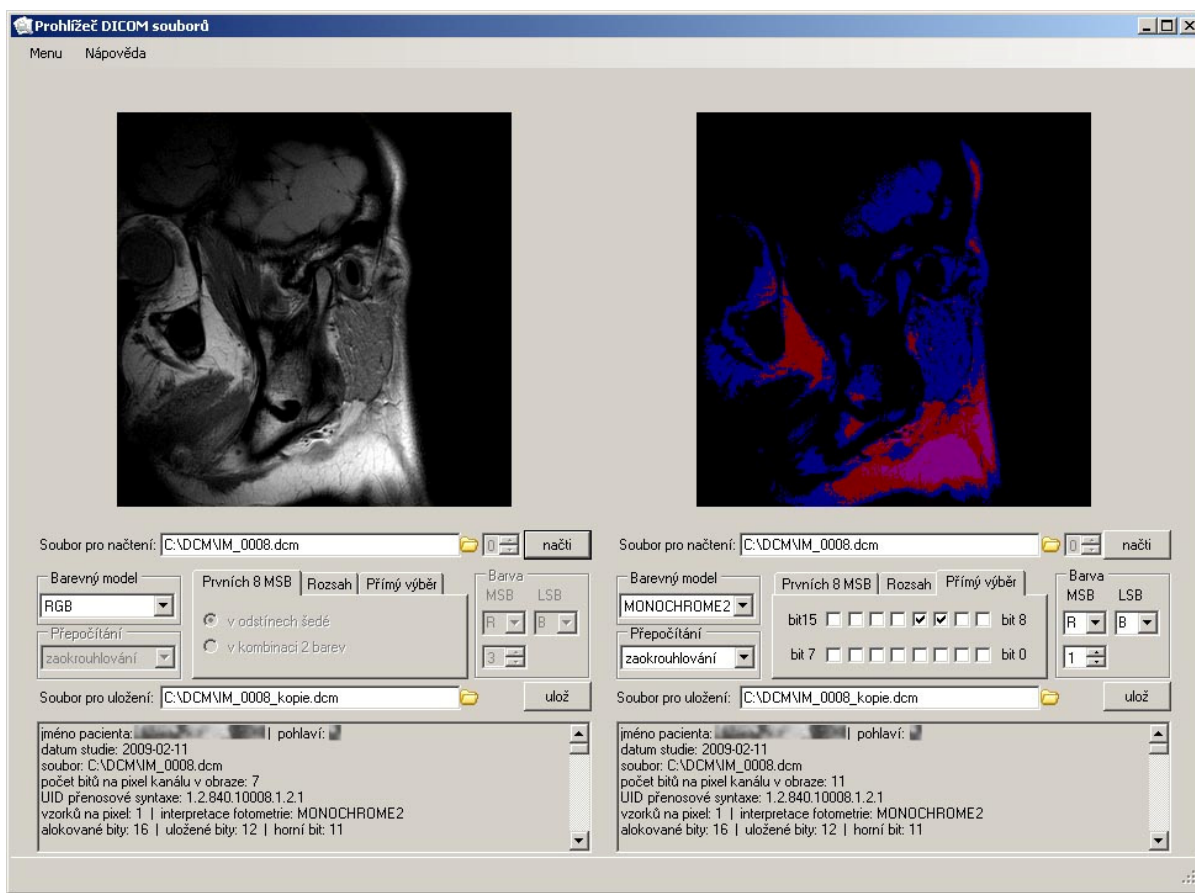
2.5 Vzhled aplikace

Grafické uživatelské rozhraní (GUI) bylo navrženo s ohledem na jednoduchost a přehlednost práce za pomoci nástavby CLR (Common Language Runtime).

Plocha je rozdělena na 2 poloviny, na sobě nezávislé části pro prohlížení obrázků. Každá se skládá z prostoru pro vykreslení obrázku, nastavení parametrů a výpisu informací.



Obr. 2.3 Nastavení parametrů



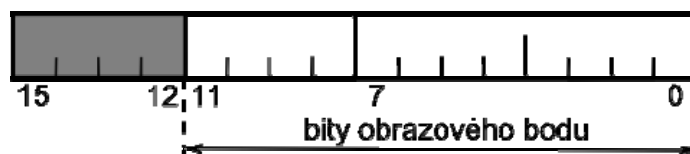
Obr. 2.4 Celkový vzhled aplikace (upraveno, aby nebyly obsaženy osobní informace pacienta)

Část pro nastavování parametrů obsahuje dvě textové pole cest souborů pro čtení a ukládání s příslušnými funkcionálními tlačítky. Mezi těmito prvky se nachází 4 bloky pro nastavení vstupních parametrů, jež ovlivňují konečný obraz:

- výběr barevného modelu:
 - RGB (barevný),
 - MONOCHROME1 (odstíny šedé),
 - MONOCHROME2 (odstíny šedé, inverzní k MONOCHROME1),
 - YBR_FULL (výhodný formát např. pro JPEG komprimaci)
- určení rozsahu zobrazovaných bitových hladin:
 - prvních 8 MSB (8 nejdůležitějších bitů)
 - rozsahem (zadání minima a maxima – LSB a MSB),
 - přímým výběrem jednotlivých hladin,
- způsob přepočítávání větších pixelů na 8 bitové:
 - zaokrouhlením (ořezáním),
 - selekcí každého druhého bitu,
- barvy při rozkladu vícebitového pixelu na dva 8 bitové:
 - výběr dvou barev (kanálů modelu RGB – červená, modrá a zelená barva),
 - nastavení počtu bitů MSB barvy (kolik horních bitů ze 16ti se přiřadí do MSB barvy, LSB barva je doplňkem do 16 bitů).

3 Výsledky práce

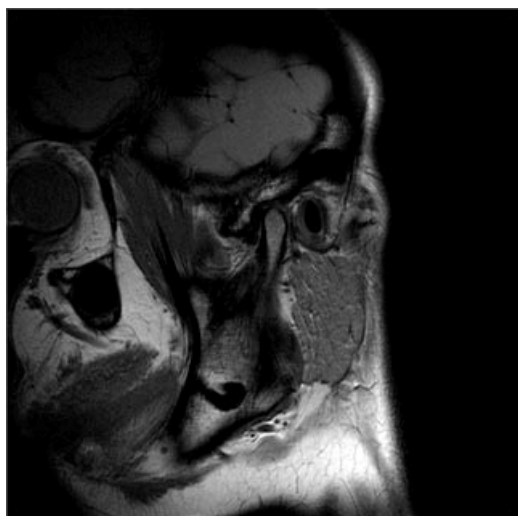
Způsoby zobrazování jsou testovány na 12 bitovém snímku hlavy pořízeného MR v rozlišení 320 x 320 bodů. V datové sadě je uložen popis interpretace fotometrie MONOCHROME2, tzn. že obrazová data mají být zobrazována v 1 kanálu škály šedi, kdy nejvyšší jasová hodnota reprezentuje bílou barvou. Je možno vyčíst, že byla použita bezeztrátová komprese. Uložené obrazové bity zvoleného studijního souboru jsou patry z následujícího Obr. 3.1.



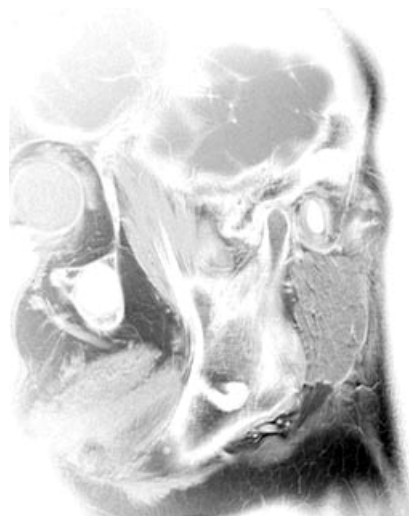
alokované bity: 16
uložené bity: 12
horní bit: 11

Obr. 3.1 Uložené obrazové bity

Načtený studijní snímek v barevných modelech MONOCHROME2 a MONOCHROME1 je vidět na Obr. 3.2 a Obr. 3.2. Zobrazení je provedeno pomocí vytvořené aplikace s definovaným základním nastavením načítání prvních 8 MSB (prvních osm bitů s nejvyšší informační hodnotou).



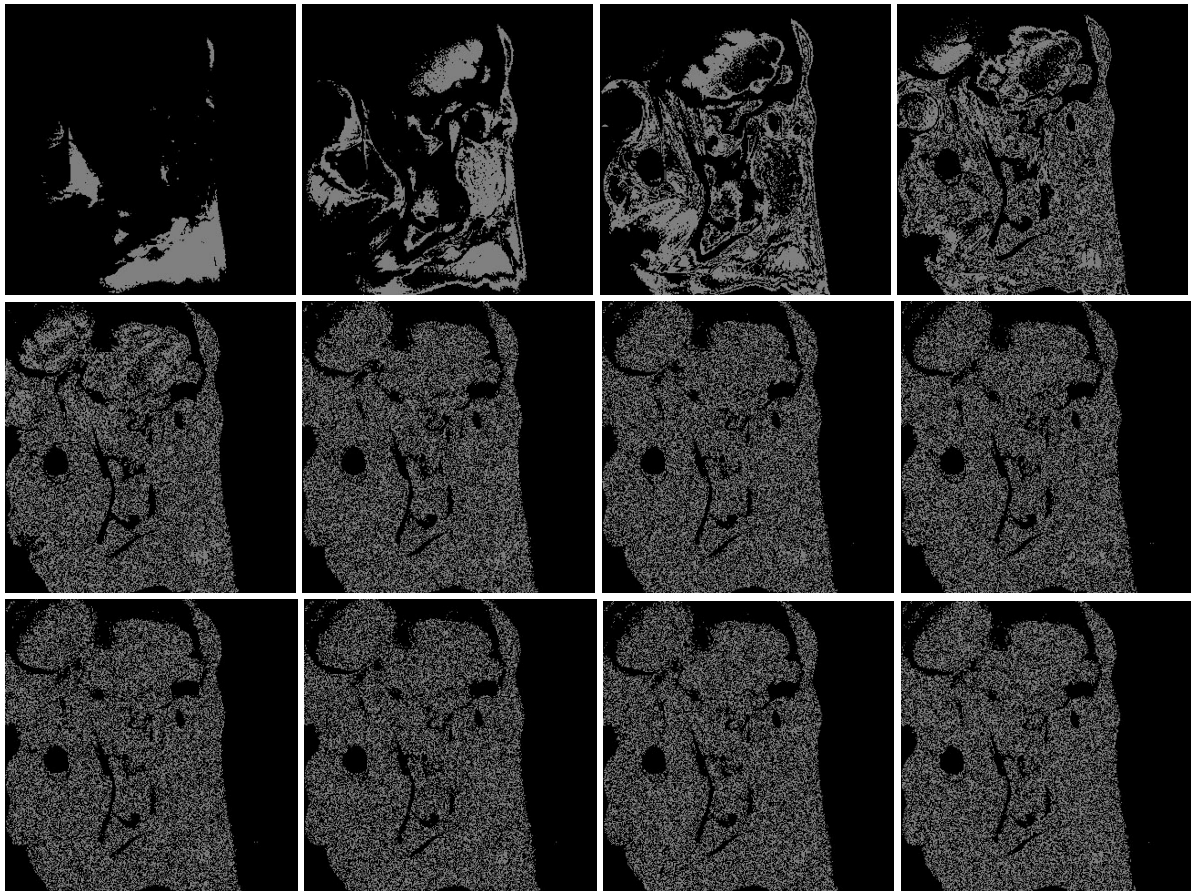
Obr. 3.2 Prvních 8 bitů MSB, MONOCHROME2



Obr. 3.3 Prvních 8 bitů MSB, MONOCHROME1

Zobrazení MONOCHROME2 je srovnatelné s většinou volně šiřitelných programů, které čtou DICOM obrazové soubory s větší bitovou hloubkou než 8 tak, že vezmou při zobrazení pouze 8 bitů s nejdůležitějšími informacemi (počínaje prvním uloženým bitem – MSB) a zbylé, méně významné, odstraní. Na předchozích dvou snímcích jsou zobrazeny bity na pozicích 11 – 4 a zbylé obrazové bity 3 – 0 odstraněny. Došlo tedy ke ztrátě nejméně čtyř hladin bitů. Pro porovnání je na Obr. 3.2 uveden barevný prostor MONOCHROME1, u kterého jsou pouze jednotlivé obrazové bity invertované, nejvyšší jasová hodnota v šedotónovém obraze reprezentuje barvu černou. Dále se budu zabývat jen daty interpretace fotometrie MONOCHROME2 z Obr. 3.2, protože v tomto barevném prostoru byla data v studijním souboru uložena.

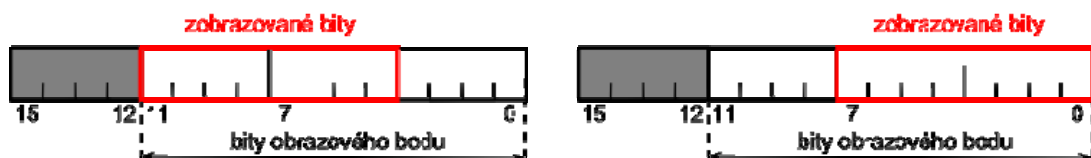
Následující sada snímků představuje vrstvy jednotlivých bitových hladin 11 – 0. Množství zanedbané informace v poslední řadě je pozorovatelné již při orientačním pohledu na rozklad originálního snímku na Obr. 3.4. Nejvíce je to patrné na části levého horního rohu.



Obr. 3.4 Rozklad do bitových hladin (vlevo nahoře bity na 11 pozici, vpravo dole na 0 pozici)

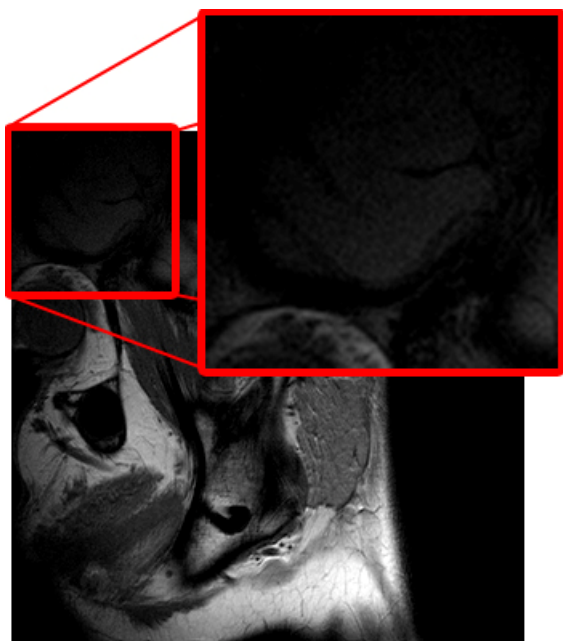
Výsledné zobrazení srovnatelné s Obr. 3.2. bychom získali sloučením prvních 8 řezů. V tomto případě je celá jedna třetina bitů představujících jemné detaily zanedbána.

Budeme-li chtít zobrazovat snímek v odstínech šedi a přesto vidět bity hladin nejnižších úrovní, můžeme posunout vybranou oblast bitů tak, jak je znázorněno na Obr. 3.5. Snímky šedé škály lze jednoduše převést do modelu RGB při zachování výsledného zobrazení tím, že jasové hodnoty šedotónového původního kanálu duplikujeme do všech třech kanálů RGB.

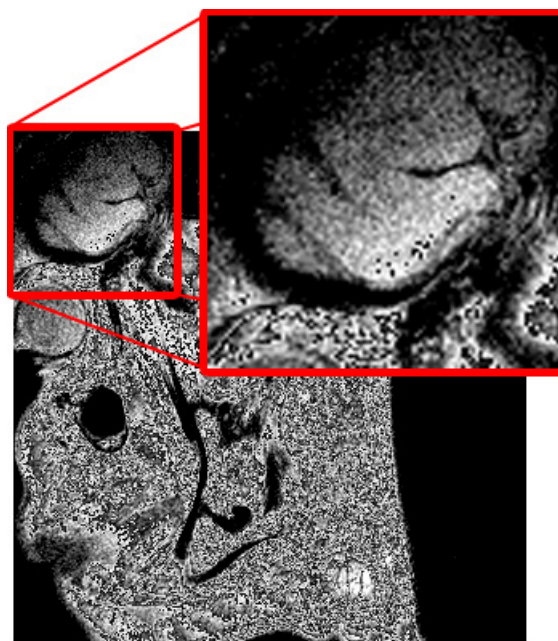


Obr. 3.5 Posun „okénka“ vybíraných bitů

Změna „okénka“ zobrazovaných bitů je vidět na Obr. 3.6 a Obr. 3.6. Ignorováním bitů s nejvyšší vypovídací hodnotou však může dojít ke ztrátě informace, jež vede k degradaci výsledného podání obrazu.



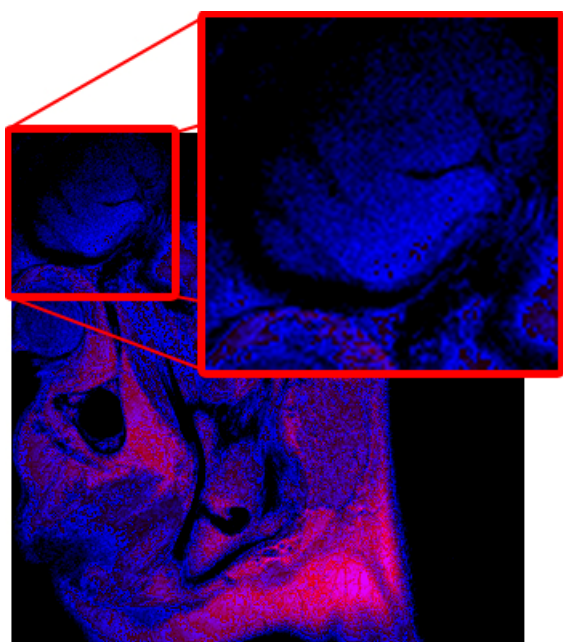
Obr. 3.6 RGB, prvních 8 MSB



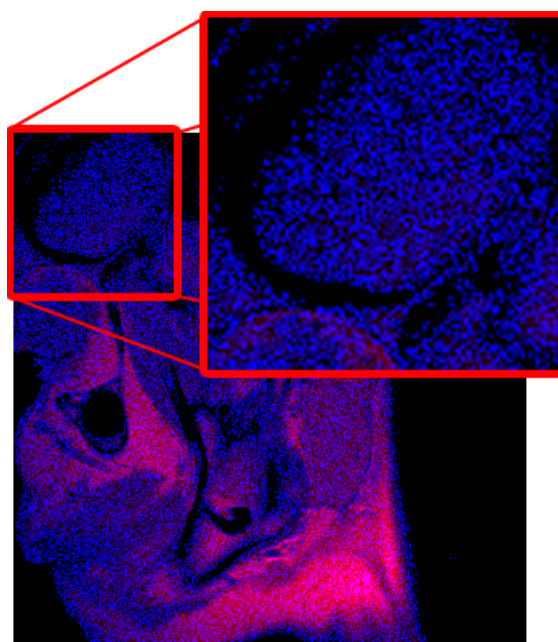
Obr. 3.7 RGB, vrstvy 8 - 0

Na dvou výše uvedených snímcích je zobrazení v modelu RGB s detailnějšími výřezy levých horních rohů, díky kterým je možné lépe oba snímky porovnat a určit jejich rozdíly. Snímek se základním nastavením vrstev 11 – 4 na Obr. 3.6 je ve vybrané části velmi tmavý a nelze jej dobře vyhodnocovat. Na dalším Obr. 3.6 byly ignorovány první čtyři hladiny, čímž se hladiny 8 – 0 dostaly více do „popředí“ (zvýšila se jasová hodnota vzorků). Daní za lepší rozeznání detailu rohu je však ztráta komplexnosti celého obrázku. Jsou znehodnocena místa, kde byly vysoké jasové hodnoty.

Aby byly informace zachovány a zobrazeny na 8 bitových zobrazovacích jednotkách, umožňuje výsledná aplikace rozdělení obrazových bitů do dvou kanálů barevného modelu RGB. Je tím sice ztraceno původní barevné podání, ale zato můžeme pomocí barev dosáhnout prezentace více informací.

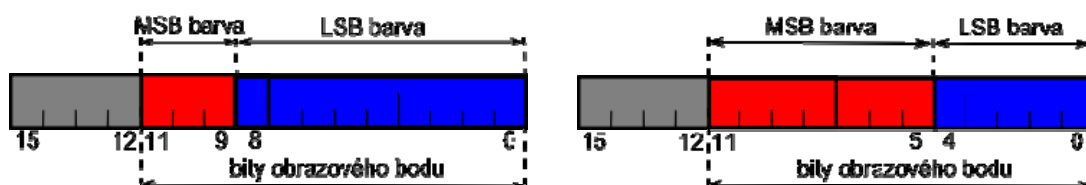


Obr. 3.8 MON_2, 11b, MSB barva R=3



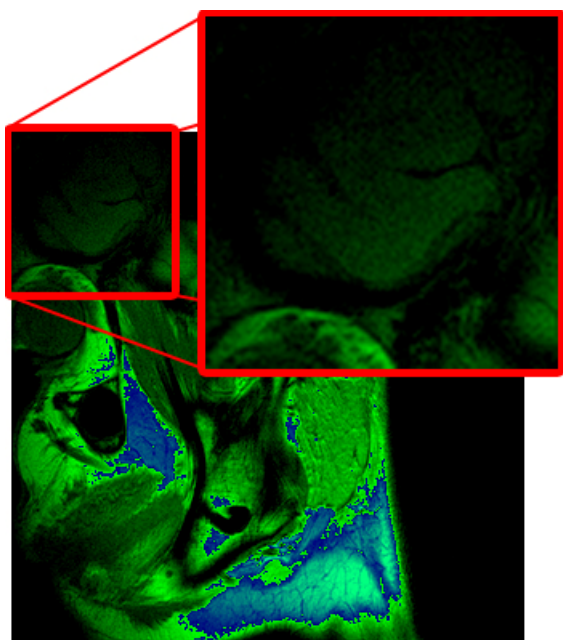
Obr. 3.9 MON_2, 11b, MSB barva R=7

Rozklad jasových pixelů barevného modelu MONOCHROME2 do 2 kanálů s plným rozsahem 11 bitů na Obr. 3.8 a Obr. 3.8 je patrný z následující ukázky. MSB barva je nastavena na červenou (R) a LSB barva na modrou (B) s přidělenými rozsahy MSB barvy (v tomto případě červené) vlevo 3 bity a vpravo 7 bitů.

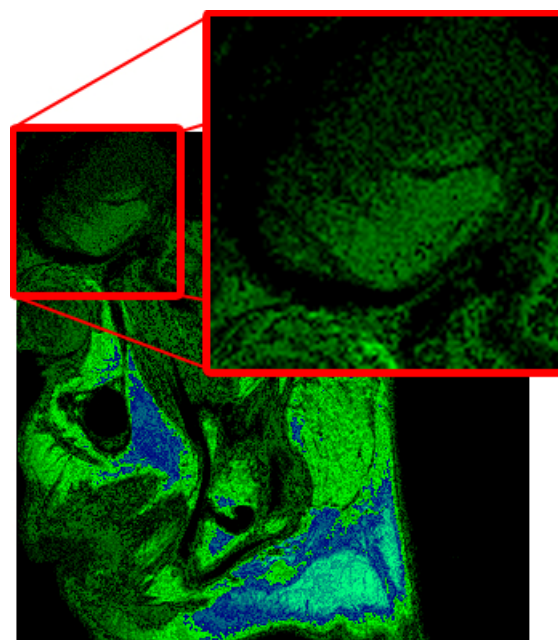


Obr. 3.10 Rozdělení bitů do 2 barev (vlevo znázorněn Obr. 3.8, vpravo Obr. 3.8)

Aplikace umožňuje více než 8 bitovou jasovou hodnotu zmenšit zaokrouhlením (de facto ořezáním nejméně podstatných informací) nebo výběrem každého druhého bitu (selekce na polovinu odstraněním sudých bitů vzorku). Přepočítání není potřebné jenom při zobrazení odstínů jedné barvy, ale i u rozkladu do dvou barev, pokud nastane situace, kdy jedna barva obsahuje více než 8 bitů.

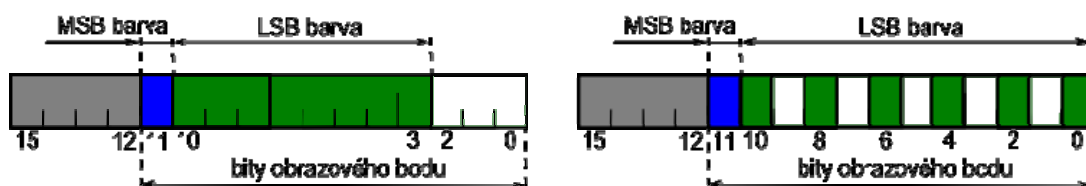


Obr. 3.11 MON_2, 11b, zaokrouhlování



Obr. 3.12 MON_2, 11b, každý druhý bit

Snímky na Obr. 3.11 a Obr. 3.11 jsou v modelu MONOCHROME2 s plným rozsahem 11 bitů. Barva MSB je modrá (B) a zachycuje jen nejvyšší hladinu. Zbýlých 11 hladin je reprezentováno zelenou barvou (G). V prvním případě vlevo jsou tyto bity na Obr. 3.11 zaokrouhleny, zobrazeny hladiny 10 – 3. V druhém případě vpravo na Obr. 3.11 je vyobrazena z bitových vrstev každá druhá, tedy bitové hladiny 10, 8, 6, 4, 2 a 0 sesunuty na 6 bitové číslo.



Obr. 3.13 Přepočítávání vzorků (vlevo zaokrouhlením, vpravo každý druhý bit)

4 Závěr

V prvním oddílu bakalářské práce jsem se snažil o přiblížení rozsáhlého standardu DICOM. Jeho hlavním cílem je sjednocení rozhraní a formátu dat, aby nebylo nutné používat v medicínských přístrojích programy pro konverzi dat mezi zařízeními od různých výrobců. Záměrem byla pouze ilustrace specifického segmentu DICOM – úložného prostoru a navazující části.

Nejdříve byly objasněny základní funkční principy a hlavní komunikační model, na němž byla směrodatná větev, která popisuje off-line komunikaci. Rysy této větve byly konkrétněji naznačeny v modelu úložného prostoru. Byla popsána struktura DICOM souboru, jejíž hlavními částmi jsou metainformace a datová sada. Jádrem standardu jsou jednotlivé prvky datové sady, které reprezentují data DICOMem přenášená nebo uchovávaná. Jsou složena z polí obsahujících tag, datovou reprezentaci, délku dat a samotná data. Samotná data obsahují jak textové informace o pacientech, tak i obrazová data. Pro kódování dat a jejich fyzickou reprezentaci využívá DICOM řady normalizovaných standardů, jejichž rozebrání nebylo náplní práce, proto zde nejsou popsány.

Vytvořená aplikace v jazyku C, C++ byla představena ve druhém oddílu bakalářské práce. Naprogramována byla za pomoci knihovny Imebra a grafické nástavby Microsoft Visual Studio 2005 – CLR. Imebra je volně dostupná knihovna pod licencí GNU GPL a poskytuje programátorovi množství tříd zpřístupňujících rozřazená DICOM data pomocí ukazatelů.

Aplikace dokáže číst DICOM soubory jejichž data lze ukládat v DICOM a JPEG formátu s možností provádět jednoduché úpravy pixelů obrazu. Program se zaměřuje na grafické znázornění až 16 bitového barevného rozložení informace DICOM obrazové modalitě na standardních 8 bitových zobrazovačích. Ve třetí kapitole jsou ukázány výsledky voleb různých barevných rozlišení, pomocí nichž je prezentován 12 bitový studijní šedotónový snímek na 8 bitovém monitoru. Dvěma zobrazovaným kanálům (barvám) je možno přiřazovat ze tří základních barev RGB modelu (červené, zelené a modré). Přepočítání větších než 8 bitových jasových informací lze buď zaokrouhlením – ve své podstatě ořezáním nejméně důležitějších informací, nebo selekcí každého druhého bitu. Pomocí těchto rozdělení do dvou barev dokáže pozorovatel rozlišit i segmenty, které jinak ztratí.

V předcházející kapitole je nejdříve ukázáno zobrazení studijního snímku tak, jak bývá předkládán většinou programů. Tyto programy dokáží prohlížet DICOM soubory zobrazením prvních 8 bitů s nejdůležitější informací, v našem případě 11 – 4 bitové hladiny. Poté byl rozsah bitů přesunut na hladiny 7 – 0, což vedlo ke zanedbání horních bitů a zvýšení jasu bodů, které byly předtím v pozadí. Pokud byla ale v pixelech hodnota jasu zasahující do odstraněných hladin, došlo ke ztrátě komplexnosti celého obrazu.

V následujícím kroku byla škála šedé barvy transformována do dvou barev, nejdůležitější (nejsvětější) bitů do červené barvy a zbývající do modré. U rozdělení 12 bitů v poměru 3 červené (R) a 9 modrých (B) bitů byla ještě v rámci možností viditelná celistvost podání zkoumaného detailu. Při rozdělení na 7 R a 5 B bitů sice byly vidět i informace v nízkých hladinách, ale zmizela jejich rozeznatelnost v celkový obraz. Díky tomuto rozdělení nastane ztráta barevného podání a celistvosti obrazu, proto je tento způsob spíše vhodný pro porovnávání jen např. 2 vrstev, nikoli detailu v obraze.

Nakonec bylo porovnáno zobrazení na studijní snímku pomocí způsobu přepočítání více než 8 bitových jasových hodnot. Nejdůležitější bitová hladina je zobrazena modře a zbylých 11 hladin je zobrazeno zelenou barvou. Nejprve byla použita metoda zaokrouhlování, kdy je možno vidět jen o jednu bitovou hladinu více než u standardní zobrazení v úvodu studie, poté byla u 11 hladin aplikována selekce každého druhého bitu. V tomto případě lze pozorovat i nejnižší bitové hladiny, jedná se o kompromis mezi ztrátou vjemu celkového obrazu a viditelností nejnižších bitových hladin.

Literatura

- [1] National Electrical Manufacturers Association: *Digital Imaging and Communications in Medicine (DICOM), Part 1: Introduction and Overview* [online], verze 2008. Poslední modifikace 26.1.2008 [citováno 3.5.2009].
Dostupné z URL: ftp://medical.nema.org/medical/dicom/2008/08_01pu.pdf
- [2] National Electrical Manufacturers Association: *Digital Imaging and Communications in Medicine (DICOM), Part 3: Information Object Definitions* [online], verze 2008. Poslední modifikace 26.1.2008 [citováno 3.5.2009].
Dostupné z URL: ftp://medical.nema.org/medical/dicom/2008/08_03pu.pdf
- [3] National Electrical Manufacturers Association: *Digital Imaging and Communications in Medicine (DICOM), Part 4: Service Class Specifications* [online], verze 2008. Poslední modifikace 26.1.2008 [citováno 3.5.2009].
Dostupné z URL: ftp://medical.nema.org/medical/dicom/2008/08_04pu.pdf
- [4] National Electrical Manufacturers Association: *Digital Imaging and Communications in Medicine (DICOM), Part 5: Data Structures and Encoding* [online], verze 2008. Poslední modifikace 26.1.2008 [citováno 3.5.2009].
Dostupné z URL: ftp://medical.nema.org/medical/dicom/2008/08_05pu.pdf
- [5] National Electrical Manufacturers Association: *Digital Imaging and Communications in Medicine (DICOM), Part 6: Data Dictionary* [online], verze 2008. Poslední modifikace 26.1.2008 [citováno 3.5.2009].
Dostupné z URL: ftp://medical.nema.org/medical/dicom/2008/08_06pu.pdf
- [6] National Electrical Manufacturers Association: *Digital Imaging and Communications in Medicine (DICOM), Part 7: Message Exchange* [online], verze 2008. Poslední modifikace 26.1.2008 [citováno 3.5.2009].
Dostupné z URL: ftp://medical.nema.org/medical/dicom/2008/08_07pu.pdf
- [7] National Electrical Manufacturers Association: *Digital Imaging and Communications in Medicine (DICOM), Part 10: Media Storage and File Format for Media Interchange* [online], verze 2008. Poslední modifikace 26.1.2008 [citováno 3.5.2009].
Dostupné z URL: ftp://medical.nema.org/medical/dicom/2008/08_10pu.pdf
- [8] National Electrical Manufacturers Association: *Digital Imaging and Communications in Medicine (DICOM), Part 11: Media Storage Application Profiles* [online], verze 2008. Poslední modifikace 26.1.2008 [citováno 3.5.2009].
Dostupné z URL: ftp://medical.nema.org/medical/dicom/2008/08_11pu.pdf
- [9] National Electrical Manufacturers Association: *Digital Imaging and Communications in Medicine (DICOM), Part 12: Media Format and Physical Media for Media Interchange* [online], verze 2008. Poslední modifikace 26.1.2008 [citováno 3.5.2009].
Dostupné z URL: ftp://medical.nema.org/medical/dicom/2008/08_12pu.pdf
- [10] National Electrical Manufacturers Association: *Digital Imaging and Communications in Medicine (DICOM), Part 15: Security and System Management Profiles* [online], verze 2008. Poslední modifikace 26.1.2008 [citováno 3.5.2009].
Dostupné z URL: ftp://medical.nema.org/medical/dicom/2008/08_15pu.pdf

- [11] MATOUŠEK L.: *Waveletová analýza a výrazňování MR tomografických a ultrazvukových obrazů*, Diplomová práce, Brno:VUT, Fakulta elektrotechniky a komunikačních technologií, 2008, 52s.
- [12] CHMELAR P, STRYKA L.: Pokročilé databázové systémy, *Multimediální databáze*, Brno: VUT, Fakulta informačních technologií, 2007 [citováno 3.5.2009].
Dostupné z URL: <http://www.fit.vutbr.cz/~chmelarp/pdb/MMDBs%200009b.pdf>
- [13] ePacs – DICOM komunikace mezi zdravotnickými zařízeními [online] [citováno 12.5.2009].
Dostupné z URL: <http://www.epacs.cz/>
- [14] Imebra – open source DICOM knihovna [online] [citováno 18. 5. 2009]
Dostupné z URL: <http://imebra.com/>

Přílohy

A Části DICOM standardu verze 2008:

PS 3.1:	Introduction and Overview	Úvod a přehled – stručný popis celého standardu.
PS 3.2:	Conformance	Komformita – definuje úroveň shody, za kterých je produkt (modalita, software) v souladu s DICOM.
PS 3.3:	Information Object Definitions	Definice informačních objektů – specifikace tříd IO, které poskytují abstraktní pohled na IO.
PS 3.4:	Service Class Specifications	Specifikace servisních tříd – servisní třídy spojující IO s příkazy vykonanými na těchto objektech.
PS 3.5:	Data Structure and Encoding	Struktura a zakódování dat – popis struktury a pravidel kódování dat v IO a servisních třídách.
PS 3.6:	Data Dictionary	Datový slovník – seznam všech datových prvků reprezentujících informaci a informační jednotku.
PS 3.7:	Message Exchange	Výměna zpráv – výměna informací mezi modalitou a okolním prostředím pomocí sdělovací techniky.
PS 3.8:	Network Communication Support for Message Exchange	Podpora síťové komunikace pro výměnu zpráv – specifikuje funkce komunikačních služeb, protokoly.
PS 3.9:	Retired	Vysloužilá část – podpora komunikace mezi 2 body pro výměnu informací, dnes v systémech nepoužitelné.
PS 3.10:	Media Storage and File Format for Data Interchange	Úložný prostor a formát souboru pro výměnu zpráv - hlavní model ukládání obrazových informací.
PS 3.11:	Media Storage Application Profiles	Aplikační profily úložného prostoru – rozšiřují třídy informačních objektů pro archivační média.
PS 3.12:	Media Formats and Physical Media for Data Interchange	Formáty médií a fyzických médií pro výměnu dat – snaží výměnu mezi aplikacemi v lékařském prostředí.
PS 3.13:	Retired	Vysloužilá část – podpora komunikace mezi 2 body pro řízení tisku, pro dnešní systémy nepoužitelné.
PS 3.14	Grayscale Standard Display Function	Norma stupnice šedi pro zobrazování – kalibrační metoda pro zobrazovací soustavu (např. monitor).
PS 3.15:	Security and System Management Profiles	Bezpečnost a profily systémového řízení – definované pomocí protokolů jako DHCP, LDAP, TLS.
PS 3.16:	Content Mapping Resource	Obsah mapovaných prostředků – vzory pro strukturu DICOM IO, kódování IO, slovník používaný DICOM.
PS 3.17:	Explanatory Information	Vysvětlující informace – informativní a normativní přílohy
PS 3.18:	Web Access to DICOM Persistent Objects (WADO)	Webový přístup do DICOM stálých objektů – žádosti pro přístup do DICOM objektů pomocí HTTP/URL.

B Obsah přiloženého CD:

- DICOM_aplikace (adresář se zdrojovými kódy výsledné aplikace)
- knihovna_Imebra (adresář se zdrojovými a hlavičkovými soubory knihovny Imebra)
- standard_DICOM_2008 (adresář s jednotlivými dokumenty standardu)
- Analyza_protokolu_DICOM.pdf (text bakalářské práce)