

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Diplomová práce

Sběr zdravotnických dat

Prohlášení

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 24. 6. 2015

Petr Žák

Abstrakt

Práce se zabývá především výměnou dat mezi různými zdravotnickými informačními systémy. Klade si za cíl v první řadě teoreticky popsat některé principy a především standardy výměny zdravotnických dat. Následně analyzovat způsob získávání, přenosu a ukládání dat z informačního systému Medicalc za účelem jejich dalšího zpracování a využití v jiných systémech, především tedy aplikace a řešení k tomuto účelu vytvořené a využívané. V poslední části pak navrhnout a zrealizovat úpravy, vedoucí k dalšímu požadovanému vylepšení a rozšíření této funkčnosti.

Abstract

This thesis focuses mainly on data exchange between different medical information systems. The first goal is to describe some principles and standards of medical data exchange in theory. Next step is to analyse ways of extracting, transporting and storing data from Medicalc information system to allow this data to be reused in different ways, while the focus is in particular on applications and solutions created and used for this purpose. The last part of this thesis proposes and then realises some changes to extend this functionality.

Poděkování

Rád bych tímto poděkoval doc. Dr. Ing Janě Klečkové za její vedení, čas a poskytnuté rady. Rovněž bych chtěl poděkovat firmě Medicalc software s.r.o, především panu Ing. Zdeňkovi Mukenšnáblovi, za skvělou spolupráci během tohoto projektu. V neposlední řadě pak patří poděkování mojí rodině za materiální i duševní podporu.

Obsah

1. Úvod.....	1
2. Data, informace a znalosti	2
2.1 Data.....	2
2.2 Informace	2
2.3 Znalosti	3
2.4 Souvislost dat, informací a znalosti	3
3. Medicínská data	4
3.1 Problematika medicínských dat obecně.....	4
3.2 Elementární data	4
4. Kvalita dat.....	5
4.1 Vnitřní kvalita	5
4.1.1 Přesnost (Accuracy).....	5
4.1.2 Jedinečnost a integrita (Uniqueness and integrity)	5
4.1.2 Objektivita (Objectivity).....	6
4.1.3 Důvěryhodnost a reputace (Believability and reputation)	6
4.2 Dostupnost	6
4.2.1 Dostupnost (Availability)	6
4.2.2 Bezpečnost (Security)	6
4.3 Kontextuální kvalita.....	7
4.3.1 Relevance a přínos (Relevancy and value added)	7
4.3.2 Včasnost (Timeliness).....	7
4.3.3 Úplnost (Completeness).....	8
4.3.4 Vhodný rozsah (Appropriate amount of data)	8
4.4 Reprezenační kvalita.....	8
4.4.1 Interoperabilita, porovnatelnost a koherence (Interoperability, comparability and coherence)	8
4.4.2 Reprezentace, srozumitelnost a interpretovatelnost (Representation, understandability and interpretability).....	9
5. Datové standardy ve zdravotnictví	10
5.1 Problematika zdravotnických datových standardů obecně.....	10
5.2 DASTA	11
5.2.1 Historie DASTA	11
5.2.2 Vlastnosti DASTA	12
5.2.3 Formát DASTA.....	12
5.3 DICOM	14
3.2.1 Vlastnosti DICOM.....	14
3.2.2 Technologie DICOM	14
5.4 HL7	15
5.4.1 HL7 Version 2 Messaging Standard	16
5.4.2 HL7 Version 3 Messaging Standard	17
5.5 Standardy pro terminologii	17
5.5.1 SNOMED.....	17
5.5.2 LOINC	18
5.5.3 UMLS	18
6. Medicalc.....	20
6.1 Klinická událost	20
6.2 Klinický proces	21
6.3 Aplikace	22

6.3.3 WinMedicalc.....	22
3.3.4 Medicalc4.....	23
7. Výměna dat v systému Medicalc	24
7.2 MEx	24
7.2.1 Principy MEx.....	24
7.2.2 Verze MEx	25
7.3 AutoExport, AutoImport.....	26
7.3.1 AutoExport.....	26
7.3.2 AutoImport.....	27
8. Analýza nových možností výměny dat systému Medicalc	28
8.1 Vlastní řešení	28
8.1.1 Návrh	28
8.1.2 Zhodnocení	29
8.2 Rozšíření MEx2	30
8.2.1 Úprava stávajících aplikací.....	30
8.2.2 Vlastní klientská aplikace	30
8.2.3 Zhodnocení	31
8.3 Rozšíření AutoExport	32
8.3.1 Zhodnocení	32
9. Analýza AutoExport	34
9.1 Uživatelské rozhraní, ovládání a princip funkčnosti	34
9.2 Použité technologie.....	35
9.2.1 Použité komponenty	36
9.3 Třídy AutoExport.....	37
9.3.1 Řídící třídy	37
9.3.2 Třídy pro zpracování dat.....	38
9.3.3 Třídy pro komunikaci, odesílání a ukládání dat	39
10. Realizace rozšíření	42
10.1 Výstupní formát	42
10.2 Implementace rozšíření.....	44
10.2.1 Třída TdmDataSetExport.....	44
10.2.2 Úpravy stávajícího kódu	45
10.3 Shrnutí současné funkčnosti	46
11. Další plán rozšíření	47
11.1 Návrh webové služby.....	47
11.1.1 Požadavky na funkčnost webové služby	47
11.1.2 Technické řešení webové služby	48
11.1.3 Rozhraní webové služby.....	49
11.2 Návrh dalšího rozšíření AutoExport.....	50
11.2.1 Třída TdmZCUSender	50
11.2.2 Úpravy stávajícího kódu	50
12. Závěr	51
Použité zdroje	52

1. Úvod

Základním účelem informačních systémů i informačních technologií obecně vždy bylo především ukládání, zpracování, přenos, získávání a prezentace dat. Od prostého uchovávání v souborech či databázích, přes jejich využívání k výpočtům či analýzám až po současné komplexní systémy pro podporu rozhodování, cílem je mít správná data, ve správný čas, na správném místě a být schopen díky nim získat požadované informace a znalosti.

Obor zdravotnictví v tomto ohledu samozřejmě není výjimkou, jelikož prakticky každá činnost a každý obor zdravotnictví, od diagnostiky po klinické studie či vývoj nových léků, generuje a zároveň vyžaduje množství různých dat. Využití informačních technologií v medicínské sféře tak samozřejmě kopíruje vývoj v jiných oblastech. Klasické informační systémy pro správu dat jsou už dnes téměř samozřejmostí a v mnoha případech se využívají moderní nástroje a systémy pro data mining, podporu rozhodování a mnoho dalších. Díky síti Internet je možné v téměř reálném čase sdílet data mezi libovolnými uživateli a systémy, což přináší mnoho možností především v oblasti spolupráce a výměny dat.

Tato práce se zabývá především výměnou dat mezi různými zdravotnickými informačními systémy. Klade si za cíl v první řadě teoreticky popsat některé principy a především standardy výměny zdravotnických dat. Následně analyzovat způsob získávání, přenosu a ukládání dat z informačního systému Medicalc za účelem jejich dalšího zpracování a využití v jiných systémech, především tedy aplikace a řešení k tomuto účelu vytvořené a využívané. V poslední části pak navrhnout a zrealizovat úpravy, vedoucí k dalšímu požadovanému vylepšení a rozšíření této funkčnosti.

2. Data, informace a znalosti

Ačkoli tyto pojmy lze považovat za všeobecně známé, vzhledem k jejich častému využívání v této práci je jistě vhodné je v první řadě přesně definovat a především vysvětlit, jaké jsou mezi nimi rozdíly.

2.1 Data

Data jsou objektivním vyjádřením faktů o reálném prostředí v podobě alfanumerických znaků, symbolů, obrazových a zvukových záznamů a podobně. [1][2] Získána jsou např. měřeními či pozorováními a reprezentují fakta o reálných objektech či procesech. [2] Jde tedy o surové údaje bez významu či souvislostí. Příkladem dat z oboru medicíny může například být:

22, 190, 75, A RH POS

Je více než zřejmé, že data je možné bez problémů ukládat, přenášet či získávat za pomoci rozličných informačních technologií, nicméně pro jejich pochopení či využití je nutno vědět, jakým způsobem je interpretovat.

2.2 Informace

Za informace lze, zkráceně řečeno, považovat *data, která mají smysl*. [3] Jedna ze známých definic informace zní:

Informace jsou data, kterým příjemce přisuzuje určitý význam na základě znalostí, zkušeností a vědomostí, kterými disponuje, a která u příjemce snižují entropii (neurčitost, neuspořádanost) vzhledem k jeho potřebám a požadavkům. [2]

Data jsou tedy nositelem informací, pokud je dokážeme správně interpretovat a využít například k získání nějakého nového poznatku. Budeme-li se držet předchozího příkladu, informace mohou vypadat takto:

Věk 22 let, výška 190cm, váha 75kg, krevní skupina A RH pozitivní

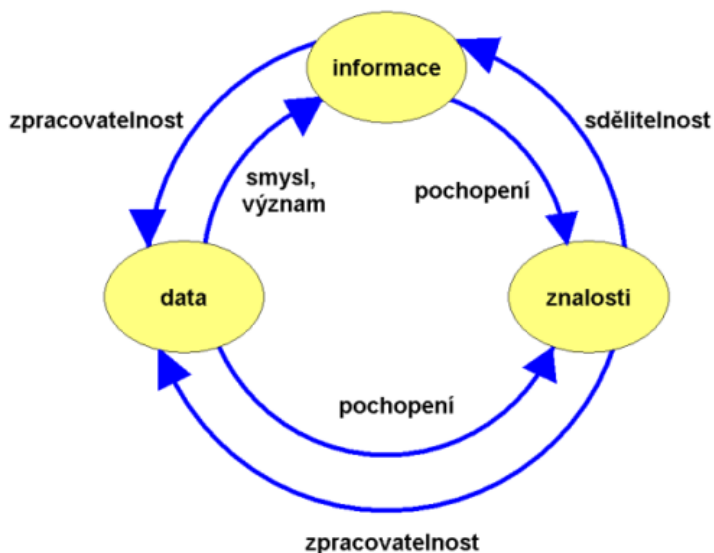
Zde je již jasné, že se pravděpodobně jedná o údaje nějakého pacienta, které může využít například jeho ošetřující lékař. Zároveň je jasně zřejmý původ pojmů informační technologie a informační systémy; jejich účelem zpravidla je ukládat, zpracovávat, přenášet a poskytovat nejen data jako taková, ale i jejich význam.

2.3 Znalosti

Znalost je informace, kterou si dokážeme začlenit do souvislostí a v kombinaci s vlastními zkušenostmi a kreativitou ji využít k získání nějaké přidané hodnoty, vyřešení problému či jiné prospěšné činnosti. [3][4] Znalost má tedy, pokud využijeme předchozího příkladu, například lékař, který ví, že dle BMI je 75kg u 190cm vysokého člověka normální hmotnost, nebo že při transfuzi může dostat i krev skupiny A negativní.

2.4 Souvislost dat, informací a znalosti

Z předchozích kapitol se může zdát, že vztah je hierarchický, tedy že informace jsou „lepší“ než data a znalosti jsou „lepší“ nežli informace. Ve skutečnosti však záleží především na použití; při plánování či rozhodování nám samozřejmě nejlépe poslouží znalosti, oproti tomu například v informačním systému je možné uchovávat jen data či informace. Vzájemný vztah asi nejlépe ilustruje obrázek 2.1:



Obr. 2.1 – vztah mezi daty, informacemi a znalostmi [2]

3. Medicínská data

Data, na jejichž základě pak vznikají potřebné informace a znalosti, jsou ve zdravotnictví, stejně jako v mnoha dalších oborech, jedním z nejzásadnějších aktiv a faktorů úspěchu. Kvalitní data pomáhají během určování diagnóz, jsou základem pro kvalifikované a správné rozhodování lékařů či jsou nezbytná pro úspěchy ve výzkumu a vývoji.

3.1 Problematika medicínských dat obecně

Zdravotnická data jsou v mnoha ohledech charakteristická. Často jsou nekompletní a liší se ve formátech, použitém jazyce či odborných termínech, způsobu zápisu a mnoha dalších ohledech. Mnoho měření a pozorování nelze opakovat, nejsou-li tedy data získána ve chvíli, kdy je to možné, nebo jsou-li následně ztracena, nelze již ztrátu nahradit. Stejně situace v medicíně často produkují různé výsledky. Mnohá z dat jsou velice citlivá a často z tohoto důvodu i legislativně chráněná. Zároveň jsou však kvalitní data v mnoha případech zcela kritická pro úspěchy jednotlivých lékařů i celých odvětví v medicíně, v případech extrémních i pro záchranu životů.

3.2 Elementární data

Nejmenší jednotkou zdravotnických dat je tzv. elementární datum, jinde též *datový element*. [12] Má jednoznačný význam a určitou hodnotu. V IT terminologii jde o objekt, který může být získán, uložen a využit v aplikaci či informačním systému. Jedná se např. o hmotnost pacienta, jeho krevní tlak, tělesnou teplotu či údaj o krevní skupině. Každé takovéto datum se zpravidla skládá ze samotné hodnoty, získané pozorováním, měřením nebo jinými postupy a dalších metadat. Těmi jsou zpravidla údaj určující, jakou vlastnost či parametr hodnota vyjadřuje, identifikační údaje pacienta a časový okamžik vzniku. [5][12]

Ve zdravotnických informačních systémech se datům často přiřazují i další metainformace či příznaky. Mezi ně patří např. údaje o tom, zda je daná informace platná trvale (alergie u pacienta) či dočasně (zvýšená teplota), nebo zda jsou data urgentní (výsledky vyšetření na JIP) či nikoli (faktura). [5]

Elementárními zdravotnickými daty jsou samozřejmě i obrazové a další záznamy pocházející z různých vyšetření (RTG, MRI, EKG apod.) či zvukové a obrazové záznamy, pocházející např. z psychologických či psychiatrických vyšetření.

4. Kvalita dat

Nejen zdravotnická, ale i veškerá další data mají řadu vlastností, které v součtu určují jejich kvalitu. Tyto vlastnosti lze jinak též označit za *metriky kvality dat*. [6] Kvalita dat do značné míry rozhoduje o jejich užitečnosti ve chvíli, kdy jsou využívána k rozhodování, výpočtům či jakékoli jiné činnosti; rozhodnutí učiněné na základě zastaralých údajů nebo výpočet provedený s nesprávnými daty s největší pravděpodobností povedou ke zcela nesprávným výsledkům. Ve zdravotnictví je na kvalitu dat přirozeně kladen veliký důraz vzhledem k tomu, že chyby mohou vést ke kritickým důsledkům.

Metriky kvality dat lze rozdělit a definovat následovně: [7][8]

4.1 Vnitřní kvalita

Vnitřní kvalita dat určuje kvalitu dat jako takových.

4.1.1 Přesnost (Accuracy)

Metrika přesnosti v sobě zahrnuje přesnost chápanou z matematického pohledu, ale především také faktickou správnost a bezchybnost dat. Zpravidla je určena např. chybovostí či mírou odchylky. [6] Např. v případě měření tělesné teploty pacienta, která je ve skutečnosti 37,08°C, není dostatečně přesný ani údaj „37,561°C“ (kterýžto má dostatečnou matematickou přesnost, ale je očividně nesprávný), ani údaj „cca 37°C“ (kterýžto je sice víceméně správný, ale naopak je nedostatečně přesný z matematického pohledu).

Ačkoli kontrola přesnosti dat je v první řadě zodpovědností toho, kdo jim dal vzniknout, informační systémy a aplikace bývají zpravidla vybaveny různými rutinami pro automatickou kontrolu dat, které snižují pravděpodobnost existence nepřesných dat. Jedná se např. o porovnávání s předem definovaným rozsahem, se staršími daty apod. [5]

4.1.2 Jedinečnost a integrita (Uniqueness and integrity)

Duplicita dat je zpravidla nežádoucí, jelikož se jedná o plýtvání zdroji a zároveň potenciální možnost porušení integrity, což je případ, kdy duplicitní data mají rozdílné hodnoty a není zřejmé, která z nich je správná. Kvalitní data mají duplicitu redukovánou na nezbytné minimum a v případech, kdy je nezbytná, nastavené mechanismy, řešící případné narušení integrity. Zpravidla se využívá určení tzv. autoritativních dat, což

jsou mezi duplicitními daty ta, která se v případě narušení integrity považují za správná. [5]

4.1.2 Objektivita (Objectivity)

Objektivní data jsou data, u nichž nedošlo ke zkreslení a žádná z jejich částí nebyla vynechána nebo zatajena. [8] Nedostatečná objektivita hrozí především u komplexnějších dat, na jejichž vzniku se podílel člověk, jako jsou např. záznamy různých pozorování či vyšetření. U dat pocházejících z přesného měření či automatických přístrojů není pravděpodobná.

4.1.3 Důvěryhodnost a reputace (Believability and reputation)

Tyto dvě metriky spolu relativně úzce souvisejí a do značné míry závisí na kvalitě zdroje dat. Důvěryhodnost je do jisté míry subjektivní a určuje, nakolik data považuje za správná a věrohodná ten, kdo je využívá či se využít chystá. Reputace pak definuje kvalitu zdroje těchto dat. Míra důvěryhodnosti a reputace dat závisí především na tom, jak kvalitní data poskytoval daný zdroj v minulosti. [8] Typickým příkladem je důvěryhodnost a reputace článků v odborných časopisech ve srovnání např. s volně editovatelnou Wikipedií.

4.2 Dostupnost

Dostupností dat je myšlena kvalita přístupu k datům.

4.2.1 Dostupnost (Availability)

Metrika, která určuje, zda jsou data k dispozici a jak náročné je získat je. Dostupnost může být ovlivněna technologickými (pomalá síť, vadné médium...), legislativními (ochrana osobních údajů) a dalšími problémy. [6] Dostupnost dat je kritická především pokud jsou potřeba k podpoře rozhodování. Například pokud se lékař, který potřebuje okamžitě podat pacientovi lék, není schopen dostat k záznamům o jeho případných alergiích kvůli chybě v informačním systému nebo pravidlům pro ochranu osobních údajů, není dostupnost dat rozhodně dostatečná.

4.2.2 Bezpečnost (Security)

Bezpečnost dat se zabývá ochranou dat před poškozením, zničením, ztrátou, odcizením a zneužitím. [9] Pro zajištění bezpečnosti dat je nutná ochrana před neočekávanými událostmi (chyby softwaru či hardwaru, poškození médií...) a cílenými

útoky (malware, hacking apod.) Mnoho dat ve zdravotnictví i jinde je velmi citlivých a proto je zaváděno mnoho mechanismů, které chrání data před ztrátou (zálohování, redundantní úložiště...) i zneužitím (šifrování, autorizace a autentizace, logování přístupů apod.) a zajišťují tak vysokou kvalitu bezpečnosti těchto dat. [5]

Úroveň bezpečnosti dat je různá v závislosti na typu a samozřejmě především důležitosti těchto dat. Například lékařské záznamy pacientů samozřejmě vyžadují zcela jinou míru ochrany než seznamy vybavení. Požadavky na bezpečnost některých zdravotnických dat bývají v mnoha případech upraveny legislativně, např. záznamy konkrétních pacientů jsou v ČR chráněny zákonem č. 101/2000 Sb., O ochraně osobních údajů a změně některých zákonů. [5]

4.3 Kontextuální kvalita

Metriky spadající do kontextuální kvality určují kvalitu dat z pohledu jejich využití k řešení problémů, rozhodování či jiné přínosné činnosti.

4.3.1 Relevance a přínos (Relevancy and value added)

Relevance říká, do jaké míry data skutečně splňují účel, ke kterému jsou určena či pro který jsou používána. [6] Relevantní data lze rovněž definovat jakožto data, která jsou pro daný účel podstatná a užitečná. Například v případě pacienta hospitalizovaného po autonehodě jsou relevantními daty jeho tep, krevní tlak, tělesná teplota, rentgenové snímky apod. Oproti tomu např. data o tom, jaké řídil auto, jsou zcela irelevantní. Přínos dat je zpravidla přímo úměrný jejich relevanci a určuje míru užitku, který díky možnosti data v dané situaci využít vznikl a množství nových informací, které byly na jejich základě získány. [8]

4.3.2 Včasnost (Timeliness)

Včasnost je definována jako *doba, za kterou lze data aktualizovat*. [6] Jinak řečeno určuje, zda data, která jsou k dispozici, jsou dostatečně aktuální. Požadavky na včasnost se samozřejmě výrazně liší v závislosti na typu dat a účelu, ke kterému jsou používána. Zatímco informace o tepu pacienta získaná na základě několik sekund starých dat již nemusí být v kritické situaci dostatečně včasná, při sestavování statistik mohou podmínku včasnosti splňovat i data stará měsíce či roky.

4.3.3 Úplnost (Completeness)

Úplnost určuje, jaká část potenciálních dat byla zachycena a je tedy k dispozici. [6] Zpravidla je nutné nalézt rozumný poměr mezi úplností a využitými zdroji, resp. omezit úplnost požadovaným rozsahem. Zvýšit úplnost je možné, v závislosti na konkrétním případě, například častějším měřením či pozorováním, ukládáním i těch dat, která byla vyhodnocena jakožto nepotřebná a podobně, nicméně pokud takto získaná data nebudou nikdy využita, je zvyšování úplnosti pouhým plýtváním zdroji.

4.3.4 Vhodný rozsah (Appropriate amount of data)

Tuto metriku lze rovněž jednoslovně označit jakožto úspornost. Vzhledem k tomu, že větší objem dat nejenže neznamena kvalitnější data, [5] nebo může kvalitu dat dokonce zhoršovat, měla by mít kvalitní data právě takový rozsah, který je dostačující pro jejich budoucí využití. Při menším rozsahu by došlo k porušení požadavku dostatečné úplnosti, zvýšení rozsahu je plýtváním zdroji a může vést k nepřehlednosti či například horší přesnosti.

Zároveň je v rámci úspornosti požadováno, aby data byla zadávána v co nejkratší a nejjednodušší formě, což znamená upřednostnění textu před obrázkem, kódu před textem apod., avšak samozřejmě pouze v případě, že nedojde ke snížení informační hodnoty. [5]

4.4 Reprezentační kvalita

Reprezentační kvalita je kvalita dat z hlediska podoby, v jaké jsou k dispozici externím uživatelům a systémům.

4.4.1 Interoperabilita, porovnatelnost a koherence (Interoperability, comparability and coherence)

Sada metrik, definujících kvalitu dat z pohledu ostatních systémů, požadujících daná data. Interoperabilita je obecně schopnost dvou systémů plně zpracovat bez jakýchkoli omezení. [10] Požadujeme-li interoperabilitu dat, je nutné je ukládat v takovém formátu a pomocí takových technologií, aby s nimi dokázaly pracovat všechny systémy, které mohou daná data potřebovat.

Porovnatelnost je *metrika hodnotící možnost porovnávat, ale také spojovat data z různých zdrojů*. [6] Míra porovnatelnosti dat závisí na použitém způsobu měření či zjišťování a především formátu ukládání dat. Neporovnatelnost dat může být způsobena

odlišným způsobem jejich získání (např. analogové vs. digitální údaje) i ukládání (např. metrický vs. imperiální systém). Zajištění porovnatelnosti dat je jedním z hlavních důvodů vývoje a zavádění datových standardů, kterými se tato práce bude později rovněž zabývat.

Již zmíněné rozdíly mezi způsobem získání dat pak popisuje koherence. Koherentní data jsou taková data, která jsou nejen porovnatelná, ale zároveň byla získána stejným způsobem, resp. za stejných podmínek a pravidel. [6] Například teploty změřené analogovým a digitálním teploměrem sice jsou vzájemně porovnatelné, ale pravděpodobně už je nelze považovat za příliš koherentní.

4.4.2 Reprezentace, srozumitelnost a interpretovatelnost (Representation, understandability and interpretability)

Druhá sada metrik reprezentační kvality dat naopak určuje, nakolik jsou čitelná a pochopitelná pro člověka, který je chce nějakým způsobem využít.

Kvalita reprezentace závisí na dvou faktorech; rozumné míře stručnosti a konzistenci zobrazení. [8] Z hlediska stručnosti je samozřejmě nutné najít rozumný kompromis, při kterém jsou k dispozici veškerá požadovaná data, ale zároveň je zachována dostatečná míra přehlednosti a čitelnosti a uživatel není přehlcen informacemi. Konzistence pak znamená, že stejná data jsou stejným způsobem poskytována; neliší se použitý způsob dodání dat, jejich zobrazení, použité jednotky apod.

Interpretovatelná data jsou taková data, která je člověk schopen „přečíst“, tedy která jsou zapsána za použití známého jazyka, standardních matematických symbolů, standardních obrazových, zvukových či video formátů, zašifrována šifrou, ke které uživatel vlastní příslušný dešifrovací klíč a podobně. Srozumitelná data jsou interpretovatelná data, která je uživatel schopen navíc pochopit a získat tak na jejich základě nové informace či znalosti. [8]

5. Datové standardy ve zdravotnictví

Datové standardy jsou dokumentované dohody definující sdílená data, jejich reprezentace a formáty. Poskytují metodu pro kodifikování získaných informací validním, smysluplným a vhodně obsáhlým způsobem. [11]

Zavedení datových standardů je nezbytné pro zajištění funkčního toku informací ve složité zdravotnické informační infrastruktuře. Díky standardům mohou být různé informační systémy integrovány do jediné infrastruktury, ve které jsou data na mnoha místech sbírána na mnoha dalších místech opětovně využívána pro rozličné účely. Datové standardy zároveň podporují zanášení nových znalostí, jako jsou např. nově zjištěné kontraindikace léků, do systémů pro podporu rozhodování. [12]

5.1 Problematika zdravotnických datových standardů obecně

V kontextu medicíny datové standardy zahrnují metody, protokoly, terminologie a specifikace pro sběr, výměnu, ukládání a získávání zdravotnických dat, jako jsou již zmiňované lékařské záznamy, informace o lécích, obrazové záznamy z vyšetření, informace o platbách apod. Základem datových standardů je standardizace datových elementů; definování dat, která se budou sbírat a způsobů, jakými budou reprezentována (určení datových typů a použité terminologie) a následovně ukládána nebo přenášena. Standardizace zpravidla zahrnuje následující: [12]

- Elementární data: Určení, jaká data budou sbírána, ukládána a šířena.
- Formáty pro výměnu dat: Definice formátů, zahrnující způsob elektronického ukládání datových prvků, postupy řešení chybových stavů, relace mezi daty či struktury používaných dokumentů.
- Terminologie: Zdravotnické termíny a koncepty používané pro popis, klasifikaci a určení datových prvků, popis vztahů mezi nimi a jazyky používané pro vyjádření těchto dat.
- Reprezentace znalostí: Metody pro elektronickou reprezentaci zdravotnické literatury, klinických postupů a obecně dat pro podporu rozhodování.

Zásadní součástí datových standardů je forma dat, definovaná pomocí datových typů. Datové typy ve zdravotnictví se zpravidla neliší od těch běžně používaných a obsahují tak jednoduché (číslo, čas, datum...) i komplexní (textový záznam...) typy. Pro

zajištění interoperability a porovnatelnosti je nezbytné použití univerzálních datových typů a rovněž jasné určení používaných jednotek.

5.2 DASTA

Datový standard Ministerstva zdravotnictví ČR, zkráceně DASTA, slouží k předávání dat mezi zdravotnickými informačními systémy, je využíván v každodenní praxi a je zabudován do všech současných významných zdravotnických informačních systémů. Jde o pravidelně aktualizovaný, otevřený standard pro komunikaci mezi informačními systémy zdravotnických zařízení, který pokrývá oblasti klinické, laboratorní, statistické i administrativní a jehož samozřejmou součástí jsou číselníky (například Národní číselník laboratorních položek, číselník klinických událostí, aktuální číselníky ÚZIS, atd.), dokumenty a nástroje (například program ČLP). DASTA je dnes v praxi využívána ve dvou základních verzích - starší DS3 a novější propracovanější verzi DS4, na kterou se soustřeďuje další rozvoj standardu. [13]

DASTA je naprosto zásadním standardem pro zdravotnická data především v ČR, mimo ČR je rovněž využíván především na Slovensku. Interoperabilita s jinými evropskými či světovými standardy v současné době neexistuje, nicméně probíhá spolupráce s pracovní skupinou standardu HL7, vzniklou v ČR a diskutuje se možná harmonizace DASTA s tímto standardem. [13][14][15]

5.2.1 Historie DASTA

Vývoj zdravotnického datového standardu pro ČR započal v roce 1992 za účasti Ministerstva zdravotnictví, lékařských fakult a několika dodavatelů informačních systémů pro zdravotnictví. Původně bylo zvažováno použití již existujících standardů, jako je EDIFACT, HL7, SNOMED, LOINC či IFCC. Po zvážení všech možností bylo rozhodnuto o vývoji vlastního řešení, které nicméně vycházelo ze zahraničních zkušeností (např. EDIFACT) a respektovalo určité požadavky, jako jsou normy ISO.

První, čistě textová verze DS 1.0, celým názvem Datový standard ministerstva zdravotnictví verze 1, byla publikována v roce 1994 a vytvořila tak základní koncept tohoto datového standardu. Již tehdy se objevilo hovorové označení DASTA. Následující vývoj byl zaměřen na rozšiřování standardu, tedy doplňování chybějících datových bloků a především rozšíření o kompatibilitu s laboratorními informačními systémy. Z tohoto důvodu začal zároveň vznikat Národní číselník laboratorních položek, zkráceně NČLP. Obě větve vývoje byly dokončeny v roce 1997, kdy byla do

praxe uvedena verze DS 1.1 zároveň s první verzí NČLP. Následně byly do roku 2001 vydány tři aktualizace a NČLP byl v deseti vlnách doplněn. Výsledná verze byla v roce 2001 oficiálně označena jako DS 1.2.

Zásadní inovací byla verze DS 2.01, která byla vydána v roce 2002 a přinesla značná rozšíření v oblasti podporovaných datových bloků, ale především byla první verzí, vytvořenou na základě standardu XML namísto čistého textu. Standard byl definován za využití DTD souborů. O rok později došlo k dalšímu upgradu, tentokrát na verzi 3.01, opět obohacenou o další datové bloky a především číselníky NZIS, tedy Národního zdravotnického informačního systému. Verze 3.01 je dodnes nejrozšířenější a nejpoužívanější verzí DASTA.

Nejnovější verzí je pak DS 4.01, která kromě novinky v podobě klinických událostí a dalších datových bloků již nevyužívá souborů DTD, ale je definována čistě pomocí XML schémat a jmenných prostorů. U většiny systémů v současnosti probíhá migrace či příprava na migraci na tuto verzi. [13][14]

5.2.2 Vlastnosti DASTA

DASTA obsahuje nejen bloky pro data týkající se pacientů, ale rovněž pro různá další laboratorní vyšetření, pro NZIS apod. Od verze DS3 standard zahrnuje mimo jiné identifikační údaje pacienta, základní informace o pacientovi (nacionále, rodné číslo, adresa, výška, hmotnost...), současné i historické diagnózy, alergie, užívané léky, očkování, anamnézu a mnoho dalšího.

Verze DS4 pak přinesla koncept tzv. klinických událostí, umožňující práci s komplexními zprávami či obecně daty pocházejícími z různých událostí; od příjmu pacienta, přes ambulantní ošetření a různá laboratorní i jiná vyšetření až po operace či propuštění pacienta. Klinické události mají pevně definovanou strukturu, sestávající se dle situace z jasně definovaných ID, typu události, informací o pacientovi, lékaři a dalších zainteresovaných osobách, zpráv o diagnóze, podaných léčích a dalších provedených úkonech, textových či obrazových výsledků vyšetření apod. [13]

5.2.3 Formát DASTA

Data reprezentovaná dle standardu DASTA jsou uložena v XML souborech s přesně definovanými názvy, skládajícími se ze znaků definujících typ dat a odesílatele, jedinečného identifikátoru souboru a přípony (např. RNN00007.xml). Rovněž hlavička souboru je přesně definována a obsahuje klasicky verzi XML, použité

kódování (povolené jsou UTF-8, IBM852, ISO-8859-2 nebo Windows-1250) a DOCTYPE, definující použití a verzi DASTA standardu a odkazující na příslušný DTD soubor.

Následuje kořenový XML element <dasta>, jehož parametry obsahují určitá metadata, jako je časové razítko, id souboru, verze použitých standardů apod. V něm jsou obsaženy další elementy, které již reprezentují jednotlivé datové bloky. Ty se pomocí parametrů rozlišují na několik typů určujících povolený obsah daného bloku (nic, data nebo další bloky), v parametrech mohou být v některých případech opět uložena metadata, jako je id, délka a podobně. Obsahem bloků, pokud je povolen, jsou pak v závislosti na jejich typu již příslušná data nebo další bloky. Standard rovněž pro každý blok definuje, zda je jeho výskyt povinný či volitelný a zda se může v rámci jednoho souboru opakovat. [15][16]

```
<?xml version="1.0" encoding="WINDOWS-1250"?>
<!DOCTYPE dasta SYSTEM "ds031001.dtd">
<dasta dat_vb="2015-06-07T16:31:23" potvrzeni="P" ozn_soub="00007" typ_odesm="NN" ur="R" bin_priloha="T" verze_ncip="02.01.01" verze_ds="03.10.01"
  <zdroj_is verze_prog="7.0.6.0" kod_prog="WMEXP" kod_firmy="MEDICALC"/>
- <pm>
  - <as typ="I">
    <vnitri/ >
  </as>
</pm>
<garant_dat odbornost="001" id_garant="01">Správce KIS Správce</garant_dat>
- <is icp="00000001" icz="00000000" ico="00000000">
  - <as typ="I">
    <obsah>Název zdr. zařízení, Ambulance</obsah>
    <vnitri>01</vnitri>
  </as>
  - <a typ="P">
    <jmeno>Název zdr. zařízení, Ambulance</jmeno>
  </a>
  - <ip id_pac="010101011">
    <rodcis>010101011</rodcis>
    <jmeno>PACIENT</jmeno>
    <prijmeni>TESTOVACÍ</prijmeni>
    <dat_dn format="D">1901-01-01</dat_dn>
    <sex>M</sex>
    <jine_idu>faktura=999</jine_idu>
  - <a typ="1">
    <jmeno>PACIENT TESTOVACÍ</jmeno>
    <adr>Plzeňská 15</adr>
    <psc>30100</psc>
    <mesto>Plzeň</mesto>
  - <as typ="T">
    <obsah>377 001 001</obsah>
  </as>
  </a>
  - <pv typ_pv="P">
    - <a typ="1" ind_kont="K">
      <jmeno>PACIENT TESTOVACÍ</jmeno>
      <adr>Plzeňská 15</adr>
      <psc>30100</psc>
    </a>
  </pv>
  - <p dat_ab="2015-06-07T16:31:23" ind_oprav_sd="N">
    <cispoj>010101011</cispoj>
    <kodpoj>999</kodpoj>
  </p>
  - <z dat_ab="2015-04-16T11:17:00" oznaceni_o="38" stav="K" obsah="NL" vznik="P" zadost="D">
    <dat_du typ="A" format="DT">2015-04-16T11:17</dat_du>
    <autor odbornost="001">Správce KIS Správce</autor>
    <nazev>001/000 - Návštěva ambulance / ordinace</nazev>
  - <text>
    <ptext xml:space="preserve">test</ptext>
    <kttext typ="text/rtf"
      encode="Base64">e1xydGYxXGFuc2lcZGVmZjB7XGZvbnc0Ym90YXZlZXQwIEFyaWFsOjE3XGZyXGZuaWxcZn
    </kttext>
  </text>
</z>
</ip>
</is>
</dasta>
```

Obr. 5.1 – Příklad XML souboru DASTA 3.1 s testovacími daty

5.3 DICOM

Digital Imaging and Communications in Medicine, v překladu digitální zobrazování a komunikace ve zdravotnictví, je standard pro zpracování, ukládání, tisk a přenos informací pocházejících ze zdravotnického zobrazování, tedy z vyšetření pomocí RTG, MRI, CT, ultrazvuku apod. V současné době je používán v mnoha nemocnicích i některých malých lékařských či např. zubařských ordinacích. [17]

3.2.1 Vlastnosti DICOM

Standard zahrnuje definici souborových formátů a komunikačního protokolu, založeného na TCP/IP. Umožňuje integraci scannerů, serverů, pracovních stanic, tiskáren a síťových zařízení od různých výrobců do tzv. PACS (picture archiving and communication system) systému, tedy systému určenému k ukládání a zprostředkování zdravotnických obrazových dat.

DICOM standard je rozdělen na dvacet souvisejících částí, které pokrývají jeho jednotlivé oblasti, jako je specifikace datových struktur, síťová komunikace, ukládání a formáty médií, způsoby zobrazování, bezpečnost či webový přístup. [17]

3.2.2 Technologie DICOM

Souborový formát DICOM je založen na datových sadách, kdy jedna datová sada obsahuje nejen příslušná obrazová data, ale i veškeré další informace, jako je např. identifikace pacienta. Datový objekt standardu DICOM se tak skládá z různých přesně definovaných atributů (datových elementů), jako je jméno či id pacienta a právě jednoho speciálního atributu, obsahujícího obrazová data. Ta mohou být tvořena jedním či několika obrázky, které je možné komprimovat pomocí různých standardů, například i klasického JPEG.

Formát souborů je však pouze jednou z oblastí, které standard upravuje. Další z nich jsou například bezpečné ukládání dat, dotazování na data a jejich opětovné získávání, automatické zjištění údajů pacienta při opětovném vyšetření, správu tisku nebo správu ukládání na vyměnitelná média. [17]

5.4 HL7

Health Level 7, zkráceně HL7, je set mezinárodních standardů pro zdravotnická a administrativní data, především pak jejich přenos a výměnu mezi systémy. Do značné míry je spíše komunikačním než datovým standardem. [19] Standard se zaměřuje na sedmou, tedy aplikační vrstvu ISO/OSI modelu; z tohoto faktu plyne i název. Primárně se skládá z následujících standardů: [18][20]

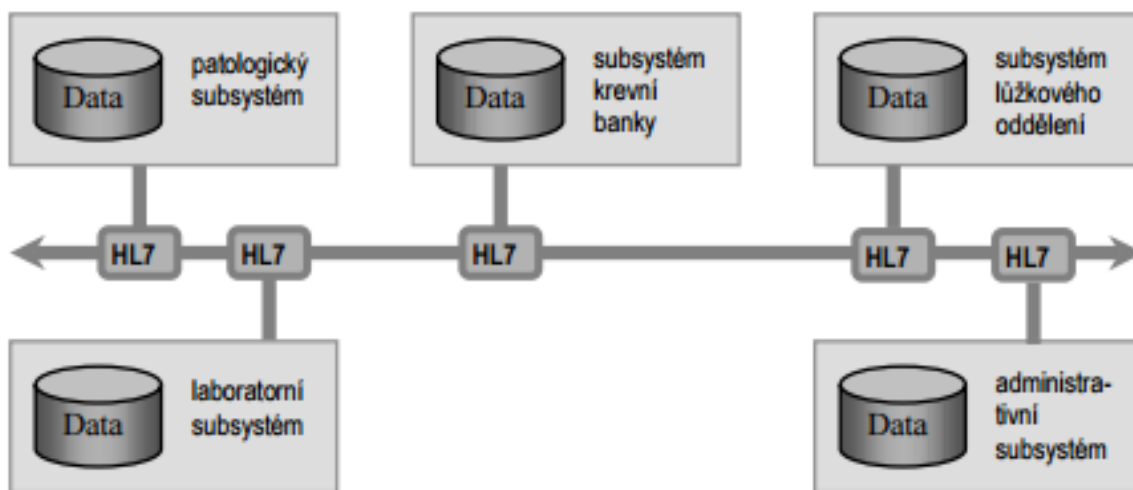
- **Version 2 Messaging Standard:** Stěžejní součást HL7; standard pro výměnu dat ve zdravotnictví.
- **Version 3 Messaging Standard:** Novější, výrazně vylepšená a změněná verze tohoto standardu.
- **Clinical Document Architecture:** Standard popisující strukturu a sémantiku zdravotnických záznamů. Je založen na XML a využívá RIM a datové typy verze 3 (viz níže).
- **Continuity of Care Document:** Specifikace pro výměnu dat o pacientech mezi lékaři v USA, založená na CDA.
- **Context Management Specifications:** Specifikace pro integraci medicínských aplikací a informačních systémů.
- **Structured Product Labelling:** Specifikace pro příbalové letáky.

Dále do této sady patří další standardy a metodologie, jako je specifikace pro elektronické lékařské záznamy, standardizace příloh dokumentům či GELLO jazyk určený pro systémy pro podporu rozhodování. [18]

Vývoj HL7 započal v roce 1987 v USA a dnes se jedná o nejrozpracovanější a celosvětově nejpoužívanější datový a komunikační standard ve zdravotnictví, zahrnující prakticky veškerá zdravotnická zařízení a oblasti medicíny. Kromě USA jej využívá dalších 25 zemí světa a to včetně řady zemí evropských; mimo jiné Německo, Polsko, Rakousko, Švýcarsko či Velká Británie. V ČR standard prakticky není využíván, pro české prostředí není přizpůsoben (neexistuje ani český překlad) a z důvodu veliké rozšířenosti nekompatibilního standardu DASTA nelze přes určité aktivity (vznik sdružení HL7 Česká republika) očekávat změnu tohoto stavu. [15]

5.4.1 HL7 Version 2 Messaging Standard

Základním cílem HL7 je *standardizovat formáty a protokoly pro výměnu datových záznamů mezi počítačovými systémy ve zdravotnictví a unifikovat tak jejich komunikační rozhraní*, jak ilustruje obrázek 5.2: [19]



Obr. 5.2 – Komunikační struktura při použití HL7

HL7 standardizuje struktury datových zpráv, jejich reprezentaci pro přenos a aplikační události, které spouštějí odeslání zpráv. Umožňuje spolupráci systémů pro administraci pacientů, praktické lékaře, laboratorních informačních systémů, informačních systémů pro lékárny a mnoha dalších. [20] Datové struktury jednotlivých systémů nejsou v HL7 standardizovány a jejich přizpůsobení je individuální záležitostí. Definovány jsou zprávy pro administraci pacientů (přijetí, přeložení a propuštění), dotazy na data, zadávání žádanek (na materiál, vyšetření...), zasílání výsledků pozorování a vyšetření, zasílání informací ohledně péče o pacienta, správu zdravotnických záznamů, upozornění na problematické stavy (lékové interakce, akutní případy...) apod. Obsah zprávy je definován formou tabulek či prostých datových typů. Standard HL7 verze 2 se postupně vyvíjel, přičemž poslední verzí je 2.6 a všechny verze jsou zpětně kompatibilní.

Datovými jednotkami ve standardu HL7 jsou zprávy, které jsou definovány vždy typem (jedinečným názvem/identifikátorem) a spouštěcí událostí. Skládají se z několika segmentů (např. záhlaví zprávy, identifikace pacienta a zpráva z vyšetření), ty mohou být povinné či volitelné a jejich opakování v rámci jedné zprávy může či nemusí být

povoleno. Segmenty se pak sestávají z políček, přičemž políčko má definovaný datový typ (číslo, datum, jméno...), je možné do něj uložit jedno elementární datum a platí pro něj v rámci segmentu stejná pravidla jako pro segmenty v rámci zprávy. Všechny prvky (zprávy, segmenty i políčka) jsou identifikována jedinečným dvoj- či třípísmenným označením.

HL7 samozřejmě řeší i bezpečnost dat při jejich přenosu, k čemuž využívá klasických bezpečnostních principů. Ověřuje autorizaci komunikujících stran a provádí jejich autentizaci, samotná komunikace je šifrována, je zajištěna integrita dat i možnost dohledat a prokázat obsah, odesílatele i příjemce veškerých přenášených dat. [19]

5.4.2 HL7 Version 3 Messaging Standard

Výhodou a zároveň i nevýhodou HL7 verze 2 je obrovská flexibilita, kdy nabízí velké množství volitelných datových segmentů i políček. Jednalo se o jeden ze zásadních faktorů úspěchu tohoto standardu, protože jej bylo možné přizpůsobit skutečně pro prakticky libovolné použití. Zároveň však tento fakt způsobil mnoho problémů s výměnou dat mezi určitými subjekty, kdy tyto používaly zcela rozdílné volitelné elementy a jejich data tak nebyla vzájemně interoperabilní i při dodržení standardu. Řešení tohoto problému je jedním z hlavních přínosů HL7 verze 3, který má metodologii tvorby a přenášení zpráv jasně definovanou pomocí tzv. Reference Information Model, zkráceně RIM. Zprávy tak mají pevně daný formát, vzájemné vztahy i životní cykly a volitelnost je ve většině případů minimální, což původní problém minimalizuje. [18][20]

5.5 Standardy pro terminologii

Zvláštní kategorií zdravotnických datových standardů jsou ty, které standardizují používanou lékařskou terminologii. Jedná se rovněž o datové standardy; použití shodné terminologie je nutnou podmínkou interoperability, porovnatelnosti a často také srozumitelnosti dat. Na rozdíl od komplexnějších standardů však neřeší formát dat, datové typy apod.

5.5.1 SNOMED

Systematized Nomenclature of Medicine, zkráceně SNOMED, je systematická, počítačově zpracovatelná kolekce lékařských a veterinárních odborných termínů. Obsahuje jasné definice kódů, termínů, synonym a definicí z oborů anatomie, nemoci,

lékařských nálezů a procedur, mikroorganismů, chemie apod. Umožňuje tak konzistenci při vytváření, ukládání, zpracování a agregaci zdravotnických dat napříč obory a zařízeními. Verze vzniklé do roku 2001 využívaly hierarchický klasifikační systém, v roce 2002 byla vydána první verze SNOMED CT (SNOMED Clinical Terms), založená na modelu „konceptů“ a vztahů mezi nimi. [21][22]

SNOMED obsahuje jedenáct základních kategorií (anatomie, bakterie a viry, léky, symptomy, diagnostika...), nazvaných osy (Axis), které pak obsahují již definované termíny, nebo další subkategorie. [21]

5.5.2 LOINC

LOINC je zkratka pro Logical Observation Identifiers Names and Codes, v překladu identifikační názvy a kódy pro logická pozorování. Jedná se o univerzální standard a databázi, přiřazující jednoznačné kódy a názvy zdravotnickým termínům, používaným při sběru a výměně výsledků klinických pozorování a laboratorních vyšetření. [23][24] Vývoj byl zahájen v roce 1994 a cílem bylo vytvořit de facto nadstavbu pro standard HL7, která zajistí, že data, vyměňovaná dle tohoto standardu, budou navíc využívat stejné identifikační kódy a názvy, což odstraní dosavadní nutnost převodu interních id jednoho systému na interní id druhého systému a dále tak usnadní výměnu dat. [25]

Formát je založen na přidělení unikátního identifikátoru každému možnému pozorování či měření. Tento identifikátor se skládá ze šesti částí, které určují subjekt, měřenou či pozorovanou vlastnost, čas, způsob měření či pozorování, měřítko či použité jednotky a použitý postup. Kromě identifikátoru má každý takovýto záznam unikátní šestimístný kód. [26]

5.5.3 UMLS

Unified Medical Language System (UMLS) vznikl sloučením mnoha zdravotnických terminologických slovníků, přičemž shodné termíny z jednotlivých slovníků jsou vzájemně propojeny mapovací strukturou, což umožňuje mezi těmito terminologickými systémy volně přecházet a překládat. Určen je především pro vývoj zdravotnických aplikací a informačních systémů, ve kterých odstraňuje či alespoň zmenšuje problém používání odlišné terminologie v různých zdrojích, což je velice užitečné při strojovém zpracování dat, vyhledávání v těchto zdrojích apod. [27]

Základem UMLS je tezaurus Metathesaurus, zahrnující přes 100 různých slovníků, mezi něž patří mimo jiné i zmiňované SNOMED CT a LOINC. Obsahuje přes milion konceptů (=objektů, entit) a přes pět milionů názvů, které různé slovníky přiřazují těmto konceptům, přičemž tato přiřazení jsou samozřejmě rovněž přenesena a z Metathesauru tak lze získávat informace o přiřazení různých termínů k různým reálným objektům. [27] Druhou hlavní součástí UMLS je sémantická síť, která přiřazuje konceptům sémantické typy (kategorie) a tyto jsou pak propojeny různými vztahy, reflektujícími reálný svět. Sémantická síť je velmi užitečná např. pro vytváření systémů pro podporu rozhodování, které jsou schopny na jejím základě vyhledávat na první pohled neviditelné vztahy a korelace. [27]

6. Medicalc

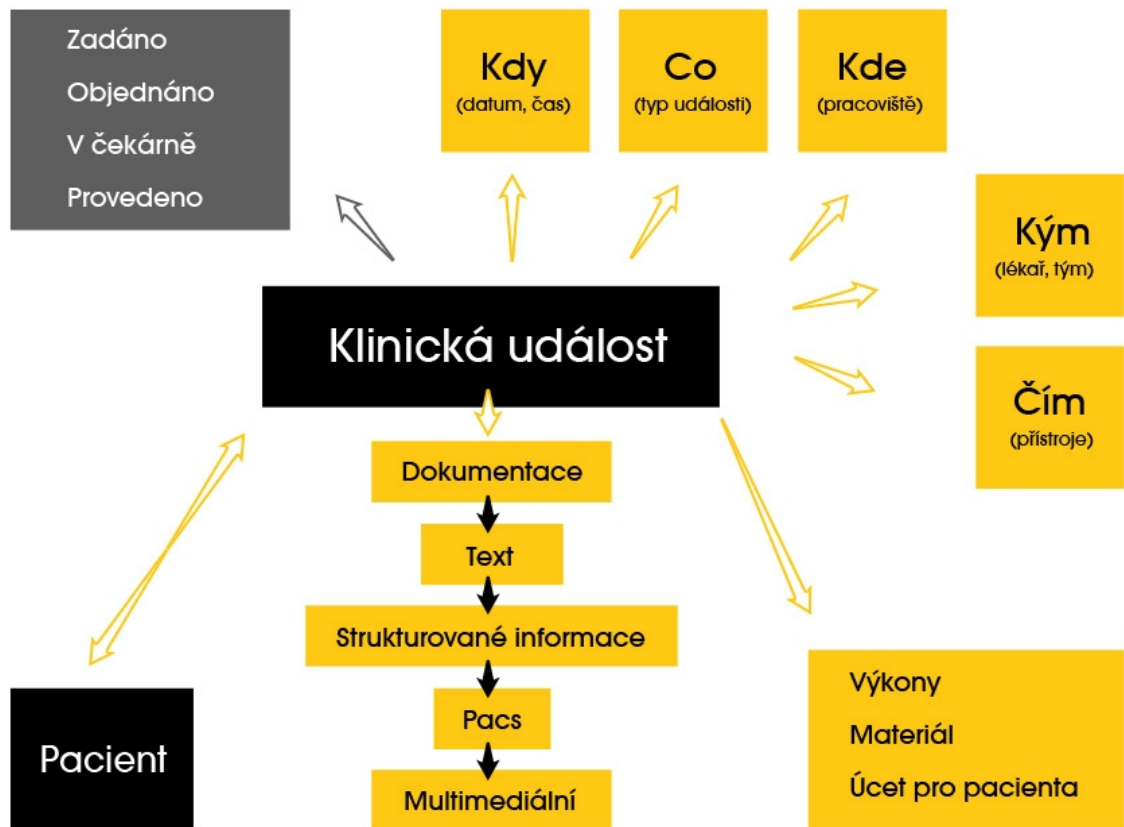
Medicalc je zdravotnický informační systém, určený pro nemocnice, vyvíjený českou společností Medicalc software s.r.o. V současné době je k dispozici čtvrtá generace s názvem Medicalc4, kterážto je přímým pokračováním předchozích tří generací produktu s názvem WinMedicalc.

Poslední, 3. verze WinMedicalc, vydaná v roce 2000, je založena na konceptu klinických událostí, který přinesl *systematický pohled na klinickou práci, harmonické sloučení textové i multimediální lékařské dokumentace a v neposlední řadě provázání s vyúčtováním zdravotní péče*. Tento koncept se následně stal základním pilířem standardu DASTA verze 4. Novější Medicalc4 pak tento koncept rozšířil o klinické procesy, čímž připravil půdu pro implementaci procesního řízení ve zdravotnických zařízeních, využívajících systém Medicalc. [28]

6.1 Klinická událost

Klinickou událostí je každá událost týkající se pacienta, kterou je možné naplánovat, objednat, provést, zdokumentovat a vyúčtovat, přičemž není podstatné, zda se odehrává ambulantně nebo v průběhu hospitalizace. Tento systém v kombinaci s centralizovaným ukládáním všech dat umožňuje v kterémkoli časovém bodě zjistit dosavadní historii pacienta a vývoj jeho onemocnění, což je užitečné zvláště v případě pacientů, na jejichž léčení se podílí více zdravotnických oborů či zařízení. Klinické události jsou zaznamenávány do jednotlivých modulů, pokrývajících veškeré možné typy takovýchto událostí (návštěva ambulance, příjem či propuštění, mnoho typů vyšetření, operace, rehabilitace, transport apod.), přičemž v závislosti na modulu se liší data, obsažená v dané klinické události.

Základem jsou samozřejmě údaje o pacientovi, datum, čas a místo, zainteresovaný lékař či lékaři atd. Dále je obsažena veškerá dokumentace, která může být představována nestrukturovaným textem, strukturovaným textem, obrazovými nebo jinými multimediálními daty. Součástí události jsou rovněž i vyúčtování nebo výkazy spotřebovaného materiálu. [28]



Obr. 6.1 – Schéma klinické události [28]

6.2 Klinický proces

Klinický proces představuje standardizované posloupnosti klinických událostí, které dohromady tvoří jeden logický proces. Může se jednat o jednoduchou posloupnost několika úkonů v rámci jednoho pracoviště, ale i o komplikovanou péči zahrnující mnoho pracovišť i zdravotnických zařízení.

Klinickým procesem je například plánovaná operace pro konkrétní onemocnění, která v první fázi znamená vyšetřování, ve druhé fázi je potvrzena diagnóza, následuje indikace k operaci, další fází jsou předoperační vyšetření, následně je realizována operace, akutní pooperační péče a nakonec navazuje ambulantní péče. Koncept klinických procesů umožňuje kdykoli získat informace o tom, v jaké fázi procesu se pacient nachází, jaké jsou výsledky již dokončených klinických událostí a které události jsou naplánovány a na kdy.

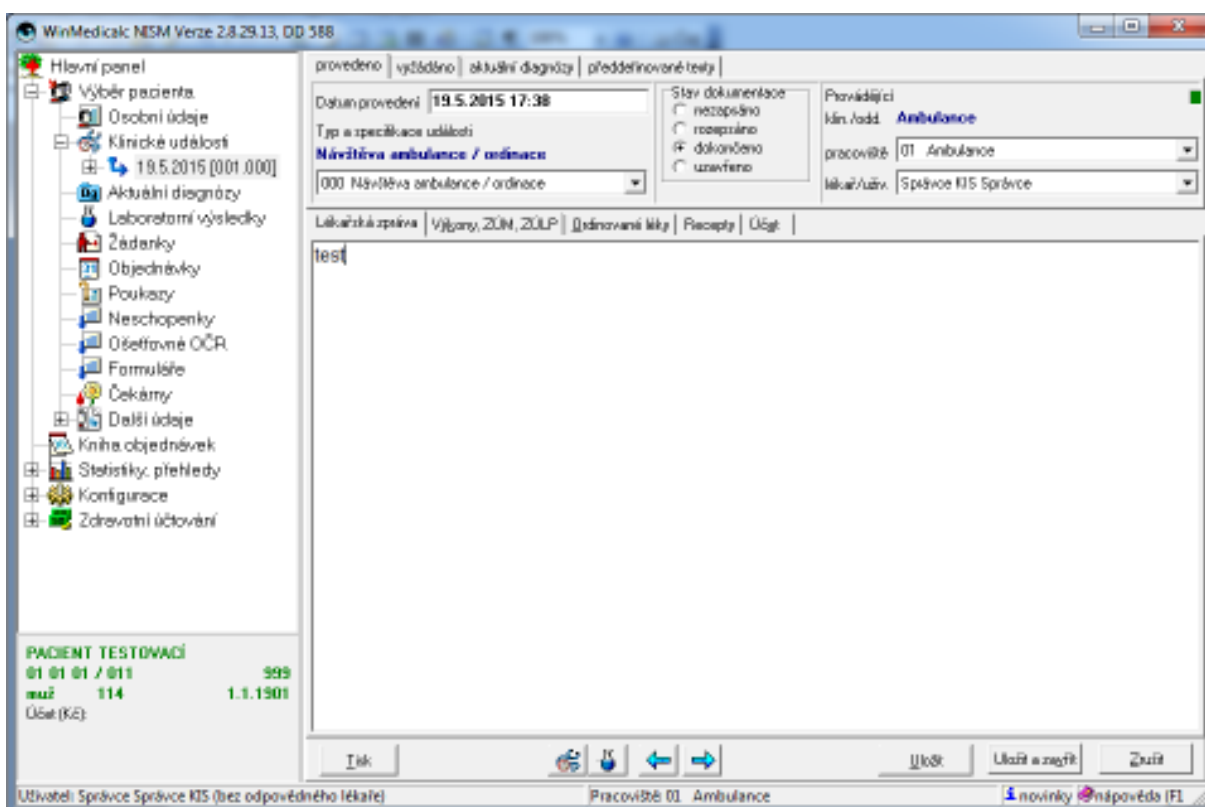
6.3 Aplikace

System Medicalc se skládá ze serverové a klientské části, přičemž obě jsou určeny pro provoz na OS Windows. Serverová část využívá databázi Oracle. Použité technologie, verze a požadavky se liší v závislosti na verzi produktu Medicalc.

6.3.3 WinMedicalc

WinMedicalc je naprogramován v jazyce C++ za využití IDE Borland C++ a Visual Component Library, přičemž využívá řadu komponent jak externích (např. Direct Oracle Access pro přístup k databázi), tak interně navržených a vytvořených přímo pro tento systém.

Klient je kompatibilní i se staršími verzemi Windows, jako jsou např. XP. Serverová část využívá databázi Oracle verze 11 a je určena pro Windows Server 2003 a vyšší, nicméně lze ji rovněž spustit i na desktopovém OS Windows. Serverová část je k dispozici v 32b i 64b verzi, klient je pouze 32b. Klientská část se ovládá standardně pomocí grafického uživatelského rozhraní, serverová část nabízí standardní i proprietární (např. SARA – SQL Administrator Rapid Access pro správu databáze) nástroje pro správu.

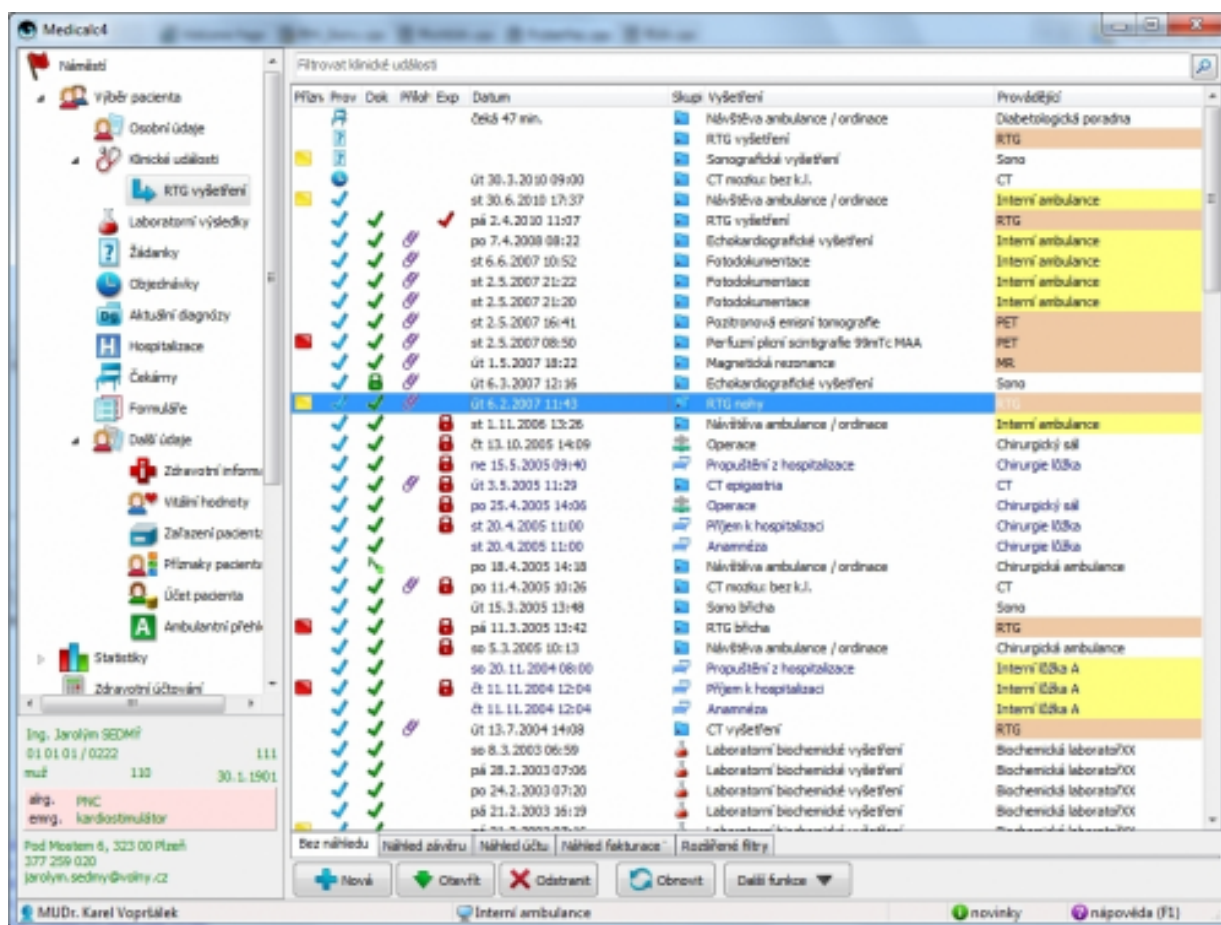


Obr. 6.2 – Grafické uživatelské rozhraní klientské aplikace systému WinMedicalc

3.3.4 Medicalc4

Architektura čtvrté generace systému Medicalc je totožná s předchozími verzemi, liší se však v použitých technologiích a v důsledku se jedná o prakticky zcela nový systém. Namísto C++ je klient i server naprogramován v jazyce C#, což znamená, že veškerá funkčnost byla implementována zcela znovu. K dispozici jsou již jen 64b verze a pro spuštění klienta je vyžadován alespoň systém Windows 7. Serverová část vyžaduje Windows Server alespoň ve verzi 2008, doporučená je však verze 2012. Verze databáze Oracle je stále 11.

Principy ovládání aplikací zůstaly víceméně totožné, pouze samozřejmě došlo k implementaci nových funkcí, modernizaci grafiky apod. Verze Medicalc4 je relativně nová a vzhledem k vysoké míře „setrvačnosti“ IT řešení ve zdravotnictví na ni většina zdravotnických zařízení zatím nemigrovala a stále tak využívají starší WinMedicalc verze 3.



Obr. 6.3 – Grafické uživatelské rozhraní klientské aplikace systému Medicalc4

7. Výměna dat v systému Medicalc

Součástí systémů Medicalc jsou řešení pro export a import dat a jejich přenos mezi různými zdravotnickými zařízeními, samozřejmě respektující standard DASTA i bezpečnostní standardy. Přenos mezi dvěma zdravotnickými zařízeními je možný jak v případě, že obě využívají Medicalc, tak v případě, že jedno z nich využívá libovolné jiné řešení, kompatibilní se standardem DASTA. Možnosti výměny dat jsou poměrně rozsáhlé a umožňují jak plně automatickou výměnu, tak manuální export i import dat.

7.2 MEx

MEx je zkratka pro Medicalc Exchange, což je primární řešení pro výměnu dat přes síť Internet při použití systému Medicalc. *Jedná se o systém pro bezpečný přenos zdravotnických dat mezi zdravotnickými zařízeními, který umožňuje výměnu lékařských zpráv, žádanek a laboratorních výsledků mezi nemocnicí a obvodními lékaři či mezi nemocnicemi navzájem. Součástí zprávy může být i obrazová dokumentace, např. snímky z provedeného RTG vyšetření.* [28]

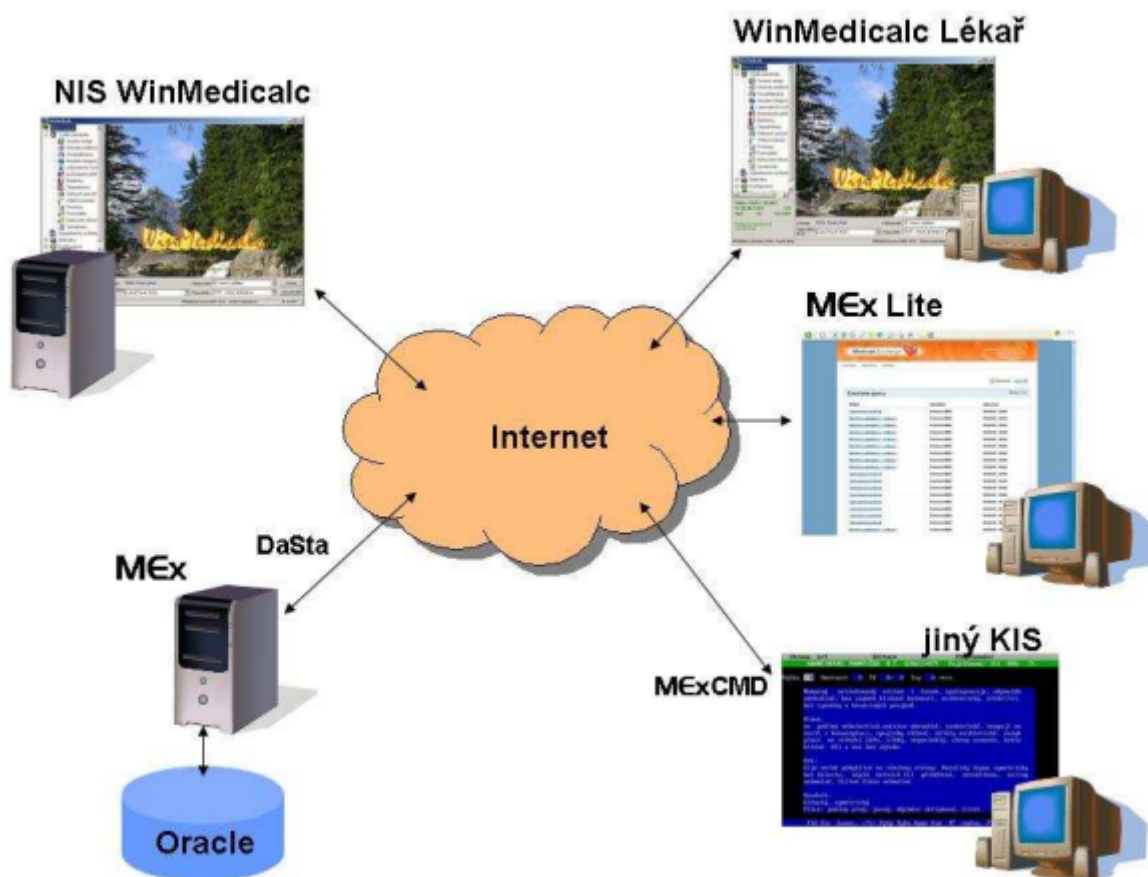
7.2.1 Principy MEx

MEx je založen na architektuře klient-server, kdy server slouží jakožto úložiště, na které jsou zasílána data z klientů a následně z něj opětovně získávána. Serverová aplikace je v principu webovou aplikací, běžící pod Apache HTTP serverem a využívající databázi Oracle.

Nejjednodušší způsob přístupu k serveru je pomocí webového rozhraní MEx Lite, které po přihlášení umožní procházet data, ke kterým má přihlášený uživatel přístup, ručně nahrávat či stahovat tato data ve formátu XML odpovídajícímu DASTA a především získat potřebné informace a certifikáty pro automatickou konfiguraci klientů. Přístup je samozřejmě bezpečný díky protokolům HTTPS a SSL. Doplnkovou funkcí webového rozhraní je možnost rezervace termínů plánovaných klinických událostí ze strany pacientů.

Zásadní funkcí je však možnost automatického připojení a následného odesílání a přijímání dat ze strany softwarových klientů systému Medicalc či jiných zdravotnických systémů za využití webové služby, se kterou je komunikováno pomocí protokolů SOAP a HTTPS. Klientské aplikace systému Medicalc mají podporu komunikace pomocí MEx samozřejmě přímo implementovanou, pro připojení klientů

jiných systémů, kompatibilních s DASTA, je k dispozici middleware v podobě aplikace MExCMD. Data jsou opět komunikována ve formátu XML, který odpovídá standardu DASTA. Komunikace je samozřejmě opět zabezpečena a šifrována, k ověření komunikujících protějšků slouží certifikáty, přičemž přístupový certifikát pro lékaře je uložen na čipové kartě, což přidává další úroveň zabezpečení.



Obr 7.1 – Schéma MEx

7.2.2 Verze MEx

K dispozici jsou dvě verze nástroje MEx; MEx první generace a MEx2, tedy MEx druhé generace. Oba stojí na stejných principech, rozdíl je v podporovaných standardech a implementaci a podpoře v koncových klientech. MEx první generace je určen pro klienty systému WinMedicalc až do verze 3, podporuje standard DASTA verze 3.1 a komunikace z klientské aplikace je řešena pomocí utilit AutoExport a AutoImport, popsané v dalších kapitolách. MEx2 je nová generace, plně integrovaná v aplikacích Medicalc4 a kompatibilní s nejnovější, 4. verzí DASTA.

MEx2 je kompatibilní i se standardem DASTA 3.1, který rovněž plně podporuje, není však již zpětně kompatibilní se staršími aplikacemi WinMedicalc verze 3 nebo nižší. Z tohoto důvodu většina zdravotnických zařízení stále používá první generaci MEx.

7.3 AutoExport, AutoImport

AutoExport a AutoImport je pár utilit, sloužících k výměně dat, konkrétně pak klinických událostí, v systémech WinMedicalc. Jak je zřejmé z názvů, AutoExport slouží k získání dat ze systému a jejich odeslání či obecně předání k dalšímu využití, AutoImport pak naopak slouží k zanesení dat získaných z externího zdroje do systému.

Obě utility jsou samostatnými aplikacemi s vlastním grafickým uživatelským rozhraním, rovněž naprogramovanými v C++ za pomoci IDE Borland C++. Na aplikaci WinMedicalc jako takové jsou nezávislé, komunikují přímo s databází systému. Spouštěny jsou na serveru, lékaři jakožto uživatelé klientských aplikací s nimi nepřijdou do styku.

7.3.1 AutoExport

Aplikace AutoExport umožňuje export klinických událostí z databáze systému WinMedicalc do různých formátů a jejich automatické odeslání různými způsoby. Výběr událostí k odeslání a jejich cíl je zvolen uživatelem v klientské aplikaci WinMedicalc, která následně veškerá data i požadované parametry exportu a odeslání uloží do speciální tabulky v databázi. Utilita AutoExport po svém spuštění na serveru získá obsah této tabulky z databáze a dávkově provede export a případně odeslání klinických událostí dle uživatelem (či uživateli) nastavených parametrů.

Veškeré parametry, jako jsou parametry databáze, servery a přístupy pro odesílání dat, používané adresáře, povolené způsoby komunikace atd. se definují pomocí .ini souboru, jenž je vytvořen dle příslušné dokumentace, která je správcům systému k dispozici.

Kombinace WinMedicalc a AutoExport nabízí následující možnosti předání textových dat klinických událostí:

- Uložení do souboru ve formátu DASTA nebo plaintextu.
- Odeslání e-mailem ve formátu DASTA nebo plaintextu.
- Odeslání na MEx server ve formátu DASTA, jak je popsáno výše.

- Odeslání do Elektronické zdravotní knížky IZIP přes bránu IZIGATE ve formátu DASTA.
- Odeslání do systému zařízení RadioMed, což je soukromé zdravotnické zařízení zaměřené na teleradiologii a telemedicínu.
- Odeslání na server služby MedicalNET, což je služba pro bezpečnou výměnu dat o pacientech, ve formátu DASTA.

Zároveň poskytuje následující možnosti předání obrazových dat klinických událostí:

- Odeslání do systému PACS kompatibilního se standardem DICOM.
- Odeslání na MEx server, jak je popsáno výše.
- Uložení do souboru.

7.3.2 AutoImport

Cílem aplikace AutoImport je dle očekávání naopak přijímat data o klinických událostech z různých zdrojů a následně je vkládat do databáze systému WinMedicalc. Principy i použité technologie jsou totožné s aplikací AutoExport. Vzhledem k tomu, že cílem této práce bylo především prozkoumat možnosti získávání dat ze systému Medicalc pro jejich následné další využití, je zde tato utilita uvedena pouze pro úplnost a práce se jí nebude hlouběji zabývat.

8. Analýza nových možností výměny dat systému Medicalc

Praktickým cílem této práce bylo prozkoumat ve spolupráci s firmou Medicalc software s.r.o. možnosti rozšíření funkčnosti exportu dat systému Medicalc o nový komunikační kanál, který by umožňoval přenos klinických událostí ze spolupracujících zdravotnických zařízení vybavených systémem Medicalc na ZČU, kde by sloužily dalšímu vývoji a případně byly předávány spolupracujícím organizacím, jako je například SITS. SITS, celým názvem Safe Implementation of Treatments in Stroke, je nezisková, nezávislá, mezinárodní výzkumná organizace, která se zabývá prevencí a léčbou mrtvice a za tímto účelem samozřejmě sbírá data o klinických případech souvisejících s tématy výzkumu.

Během úvodní analýzy a konzultací s firmou Medicalc software s.r.o. bylo navrženo a prodiskutováno několik možných řešení.

8.1 Vlastní řešení

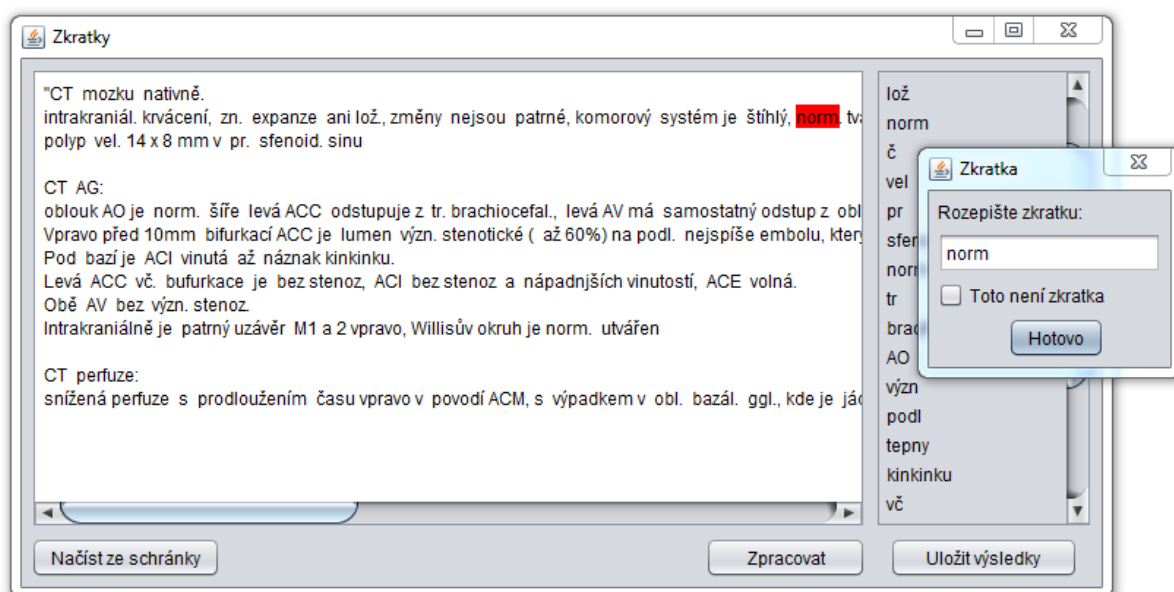
Původní úvahy směřovaly k vývoji vlastního řešení v podobě externí Java aplikace pro lékaře. Cílem bylo rozšířit funkčnost nejen o nový kanál pro získávání již existujících dat, ale i o sběr zcela nových dat od lékařů, ochotných spolupracovat na vývoji. Konkrétně se jednalo o používané zkratky odborných termínů, rozepsané do plné délky. Dostatečné množství takovýchto dat by bylo velmi užitečné při vývoji aplikace na automatické rozepisování zkratk, jejíž vývoj paralelně probíhal.

8.1.1 Návrh

Filosofie tohoto řešení spočívá v tom, že lékař ve volných chvílích vždy rozepíše zkratky u zprávy, kterou právě četl, dopsal či s ní jinak pracoval. Získaná data budou okamžitě určeným způsobem (plán počítal se zabezpečeným přenosem a ukládáním do vzdálené databáze) automaticky předána k dalšímu zpracování mimo tuto aplikaci. Z technického hlediska se jedná o Java aplikaci, která vždy umožní rychlé načtení jedné zprávy, detekci a rozepsání zkratk a odeslání získaných výsledků. Byl kladen důraz na rychlost, uživatelskou přívětivost a jednoduchost.

Program je ovládán přes grafické rozhraní, umožňuje plně nesequenční práci se zprávami, je velmi rychlý a používá moderní a přehlednou grafiku. Text zprávy je možno vepsat ručně či vložit ze schránky, ať už klávesovou zkratkou, nebo pomocí

příslušného tlačítka v aplikaci. Text ze schránky se rovněž do okna vloží automaticky při spuštění aplikace. Klepnutím na tlačítko Zpracovat se provede detekce zkratek za použití regulárních výrazů. Všechny nalezené zkratky jsou následně vypsané do menu v pravé části okna. Při kliknutí na libovolnou položku se zobrazí dialogové okno ve kterém je možné rozepsat zkratku či zvolit, že se o zkratku nejedná. Příslušná zkratka je zároveň zvýrazněna v okně se zprávou. Mezi zkratkami je možno libovolně přecházet, opakovaně se k nim vracet, měnit již zapsaná data či provést úplně novou detekci zkratek nad upravenou či zcela odlišnou zprávou.



Obr. 8.1 – Ukázka GUI prvního navrženého řešení

8.1.2 Zhodnocení

Jak je zřejmé, v počátcích práce nebylo o jiném směru vůbec uvažováno a tento návrh řešení tak byl částečně rozpracován, nicméně po dalších analýzách a diskusích bylo rozhodnuto o ukončení vývoje a určité změně směru této práce. Důvodem bylo především to, že širší nasazení a používání takovéto aplikace ve zdravotnických zařízeních se ukázalo jakožto problematické. Důvodem byly politiky IT v daných zařízeních a vytíženost lékařů, kteří by museli část svého času dobrovolně věnovat přípravě a odesílání dat, v důsledku čehož by pravděpodobně většina potenciálních dat nebyla nikdy zpracována a odeslána.

8.2 Rozšíření MEx2

První možnost, navržená ve spolupráci s firmou Medicalc software s.r.o, byla úprava systému MEx druhé generace tak, aby data odesílaná pomocí tohoto systému mohla být ve vybraných případech zároveň automaticky dodávána i na ZČU. Nabízelo se samozřejmě několik řešení; úprava kódu klienta MEx2 v aplikaci Medicalc4, úprava serverové části MEx2 nebo vytvoření speciální klientské aplikace, která by sbírala data ze serverů spolupracujících zařízení. Zde již byla opuštěna původní úvaha zahrnující doplňování dat na straně odesílající organizace, cílem bylo pouze vytvoření způsobu získávání již existujících dat.

8.2.1 Úprava stávajících aplikací

Úprava stávajícího kódu systému MEx2 by znamenala doplnění o funkčnost, která by požadovaná data exportovala v navrženém formátu a následně je zvoleným způsobem (pravděpodobně zabezpečené spojení přes Internet, případně XML soubory zasílané e-mailem či přenášené na fyzických médiích) dodávala na ZČU.

Výhody tohoto řešení byly následující:

- Použití moderního jazyka a obecně prostředí C#.
- Vzhledem k podpoře nejnovějších verzí systémů Medicalc by bylo dlouhodobě funkční, udržitelné a použitelné.
- Možnost plné automatizace a tedy praktické nezávislosti (s výjimkou prvotního schválení a konfigurace) na IT či podpoře organizace.

Za nevýhody pak lze považovat:

- Možné potíže s bezpečnostními politikami a legislativou (de facto by se jednalo o „zadní vrátka“ k datům)
- Dosavadní malá rozšířenost Medicalc4 a MEx2, přičemž zlepšení tohoto stavu v dohledné době nelze předpokládat. V důsledku tedy malé množství získaných dat.

8.2.2 Vlastní klientská aplikace

Teoretická možnost, která by využívala již existujícího komunikačního rozhraní služby MEx2. Cílem by bylo navržení a implementace nové klientské aplikace, nezávislé na Medicalc4, běžící na straně ZČU. Ze strany spolupracujících zařízení by

byly vydány přístupové údaje a certifikáty, které by byly využity k přístupu pomocí webové služby přes protokol SOAP a stažení požadovaných dat z MEx2 serveru. Aplikace by zároveň mohla sloužit k dalšímu zpracování dat, exportu do požadovaných formátů, předání do jiných systémů apod.

Toto řešení by nabízelo následující výhody:

- Technologickou a platformní nezávislost, klientská aplikace by mohla být implementována v libovolné podobě – C# či Java desktop aplikace, webová aplikace apod.
- Bezpečnost; byly by použity veškeré bezpečnostní standardy systému MEx2, jako je autorizace ze strany zdravotnického zařízení, autentizace pomocí hesla a čipové karty, šifrované spojení...
- Stejně jako v předchozím případě dlouhodobá udržitelnost, funkčnost a podpora díky spolupráci s nejnovější verzí systému MEx2.
- Celková nezávislost na implementaci systémů Medicalc.

Nevýhody by byly následující:

- Nemožnost definovat stahovaná data a jejich formát.
- Nutnost spravovat přihlašovací údaje a certifikáty, nutnost použití čipových karet.
- Stejně jako v předchozím případě malá rozšířenost systému MEx2 a z toho plynoucí malé množství získaných dat.

8.2.3 Zhodnocení

Využití MEx2 by bylo moderním řešením s velkým potenciálem pro budoucí použití i další vývoj, nicméně po dohodě s firmou Medicalc software s.r.o. bylo rozhodnuto o tom, že zatím nebude implementováno. Hlavním důvodem pro toto rozhodnutí byla především společná nevýhoda obou jeho variant, kterou je nízké tempo migrace na systémy Medicalc4 a MEx2 ze strany zdravotnických zařízení.

Lze však předpokládat, že před ukončením podpory starších verzí WinMedicalc a MEx první generace bude nutné tuto funkčnost zprovoznit i v nových verzích a tato analýza pak může posloužit jako návrh řešení.

8.3 Rozšíření AutoExport

Využití utility AutoExport bylo druhou uvažovanou možností. Jednalo se o možnost poměrně přímočarou, spočívající v rozšíření této utility o další formát a způsob komunikace, určené právě pro předávání dat na ZČU. Následné diskuse s dodavatelem Medicalc vedly k rozšíření tohoto návrhu na vytvoření obecného formátu, který bude umožňovat získání libovolných dat z databáze systému Medicalc a jejich export do souboru či odeslání určeným způsobem. Původně požadovaný přenos klinických událostí na ZČU tedy v důsledku je pouze jedním z možných využití tohoto řešení, které může být v závislosti na zvolených vstupních datech v budoucnu mnohokrát opakovaně využito pro další účely.

Oproti tomu byl i zde opuštěn plán na získávání dodatečných dat a cílem tedy opět byl pouze export dat již obsažených v databázi. Rovněž bylo nutné brát v potaz to, že podpora utility AutoExport končí s WinMedicalc 3 a v zařízeních, která již používají Medicalc4, tak nebude možné toto řešení nasadit.

Výhody tohoto přístupu jsou zřejmé:

- Univerzálnost a opakovaná použitelnost, vyhovující dodavateli systému Medicalc.
- Relativně jednoduchá implementace do stávající aplikace, možnost využít již hotových částí této aplikace.
- Podpora v současné době nejrozšířenější verze WinMedicalc 3 a z toho vyplývající možnost rychlého nasazení v řadě různých zařízení a okamžitého zahájení pilotního provozu získávání dat.

Stejně tak i zásadní nevýhoda:

- Nekompatibilita s nejnovější verzí aplikace a samozřejmě ani s případnými budoucími verzemi a z toho plynoucí nutnost na konci životního cyklu WinMedicalc 3 problém znovu řešit jiným způsobem.

8.3.1 Zhodnocení

Na základě dohody s firmou Medicalc software s.r.o. byl tento způsob řešení vybrán k realizaci. Vzhledem k poměrně bezproblémové implementaci a širokým možnostem využití se implementace rozhodně vyplatí i přesto, že provoz utility AutoExport bude ukončen společně s WinMedicalc. Tento přístup vyhovuje rovněž i

ZČU vzhledem k tomu, že z něj plyne možnost nasazení za velmi krátkou dobu a do mnoha zařízení, což zajistí přísun dostatečného množství dat v dostatečně blízké budoucnosti.

Na konci životního cyklu produktu WinMedicalc 3 bude samozřejmě nutné včas vyřešit náhradu tohoto řešení. Rozhodně lze doporučit využití MEx2, které bylo navrženo v předchozí kapitole.

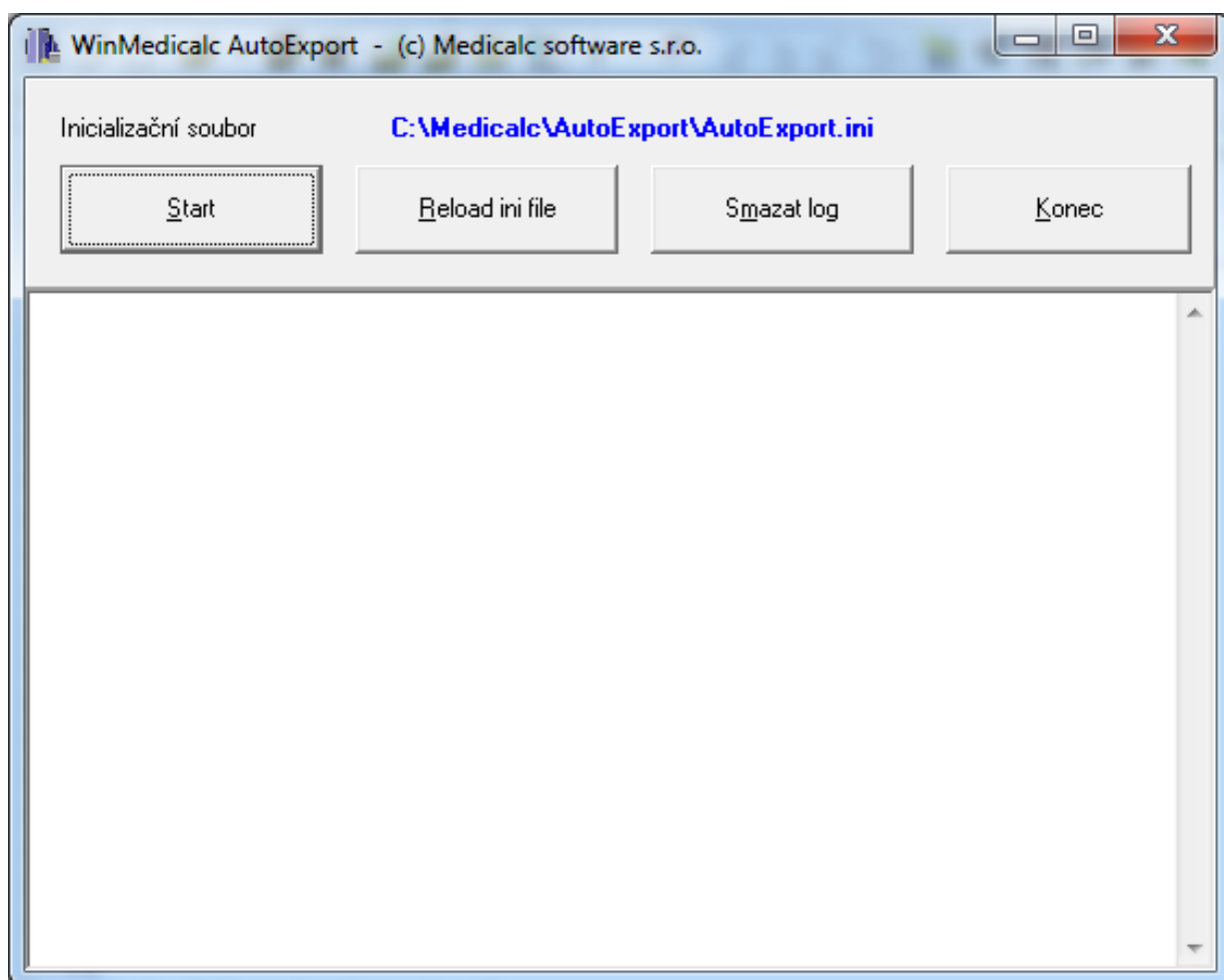
9. Analýza AutoExport

Pochopení stávající aplikace AutoExport do té míry, aby bylo možné ji rozšířit o požadovanou funkčnost, se vzhledem k absenci dokumentace ukázalo být netriviálním a přitom samozřejmě poměrně zásadním úkolem. Veškeré získané poznatky jsou uvedeny v této kapitole a v případě potřeby, například při dalším rozšiřování funkčnosti o export jiných dat, mohou do jisté míry nahradit chybějící dokumentaci.

9.1 Uživatelské rozhraní, ovládání a princip funkčnosti

Aplikace má jednoduchou podobu .exe souboru. Veškeré nastavení se provádí za pomoci .ini souboru, ve kterém je možné definovat logování, kódovou stránku, povolené druhy komunikace, přístupy do databáze, parametry komunikace s veškerými servery, na které jsou data odesílána apod. Rovněž je zde možné definovat adresáře, do kterých budou ukládány výstupy aplikace – soubory s exportovanými daty, soubory s logy a hlášeními o chybách a dočasné soubory.

K ovládání aplikace následně slouží jednoduché GUI, které slouží prakticky pouze ke spuštění dávkového zpracování dat určených k exportu a zároveň v textové konzoli zobrazuje stavové a případně chybové informace. Data určená k exportu jsou, jak již bylo zmíněno, předem zvolena uživatelem v aplikaci WinMedicalc. Ta je následně společně s uživatelem nastavenými parametry exportu uloží do speciální tabulky (LG_EXTZAR_EXPORT) v databázi systému, odkud jsou po spuštění procesu aplikací získána, zpracována a při úspěšném průběhu následně vymazána.



Obr 9.1 – GUI aplikace AutoExport

9.2 Použité technologie

Aplikace je, jak už bylo zmíněno, vytvořena v jazyce C++ za použití IDE Borland C++. Zároveň je de facto od základu postavena na Visual Component Library (VCL), což je objektově orientovaný framework původně určený pro tvorbu uživatelských rozhraní pro Microsoft Windows aplikace vytvářené v C++. Jak je zřejmé již z názvu, je založen na tzv. komponentách, což jsou předpřipravené, znovupoužitelné třídy poskytující úzce specifikovanou, konkrétní funkčnost. Zásadní výhodou VCL, využívanou v aplikacích AutoExport i WinMedicalc, je fakt, že komponenty nemusejí být pouze vizuální (určené pro použití v GUI), ale prakticky libovolné. Díky tomu existuje mnoho komponent a sad komponent pro přístup k databázím, síťovou komunikaci, práci s XML apod. Další výhodou je možnost samozřejmě navrhovat a implementovat vlastní komponenty a např. pro WinMedicalc tak existuje široká sada proprietárních komponent.

9.2.1 Použité komponenty

AutoExport používá celou řadu VCL komponent. Použity jsou především komponenty z balíků Direct Oracle Access, The Indy Project (komponenty pro síťovou komunikaci), ICE XML a samozřejmě WinMedicalc components. Následuje popis několika zásadních komponent, které byly nebo mohou být použity při implementaci rozšíření aplikace:

- **TDataModule:** Kontejner pro správu nevizuálních komponent, zpravidla používaných pro práci s daty. Většina tříd aplikace je potomkem této komponenty. Při návrhu aplikace v IDE de facto nahrazuje klasický formulář, používaný u vizuálních komponent a umožňuje pracovat s nevizuálními komponentami stejným způsobem; po přetažení komponenty do kontejneru automaticky deklaruje potřebné proměnné, vloží hlavičkové soubory apod., nastavení jejich atributů je možné provádět rovněž přímo pomocí IDE atp.
- **TMultiQuery:** Proprietární komponenta vytvořená pro Medicalc děděním z **TOracleDataset**. Jedná o komponenty pro práci s databázovými daty, umožňující na základě SQL dotazu a vytvořeného spojení do databáze získat data, uložit je a zprostředkovat k nim přístup pro jejich využití v aplikaci.
- **TMultiPartText:** Další specializace **TMultiQuery**, určená pro práci se složitějšími textovými daty.
- **TMultiPackage:** Rozšíření **TOraclePackage**, což je reprezentace Oracle balíčků PL/SQL funkcí, procedur apod., umožňující používat obsah těchto balíčků z programu.
- **TOracleQuery:** Jednoduchý dotaz do databáze Oracle, napsaný klasicky v jazyce SQL a spuštěný přes **TOracleSession**.
- **TOracleSession:** Komponenta představující spojení do databáze, po nastavení parametrů a přihlášení zprostředkuje spouštění dotazů, kontrolu transakcí apod.
- **TICEXMLComponent:** Objekt představující XML dokument a nabízející funkčnost pro konstruování tohoto dokumentu, jako je vytváření, přesouvání a mazání elementů v hierarchii, export dokumentu v podobě řetězce apod.
- **SAXParser:** Tato komponenta je určena k parsování XML dokumentů.

- **TipsSOAPS:** Komponenta pro komunikaci s webovými službami přes protokol SOAP, podporující rozšíření Web Service Security (WS-Security, WSS) pro zabezpečení.
- **TipsHTTPS:** Komponenta pro komunikaci přes protokol HTTPS.
- **TipsCertMgr:** Komponenta pro práci s úložištěm certifikátů systému windows, umožňující přístup k certifikátům pro zabezpečenou komunikaci.
- **TipmSFileMailer:** Umožňuje zkonstruovat hlavičku, tělo i přílohu e-mailové zprávy a odeslat ji za pomoci SMTP serveru určeného na základě parametrů. Rovněž podporuje šifrování.

9.3 Třídy *AutoExport*

Původní aplikace je složena z 22 tříd. Tato práce si neklade za cíl plnohodnotně nahradit dokumentaci k aplikaci a proto je popis spíše zběžný, popisující jejich funkčnost a upozorňující na důležitá fakta a nejzásadnější metody.

9.3.1 Řídící třídy

Třídy, které jsou určeny pro ovládání a řízení běhu aplikace a dále třídy, zajišťující různou podpůrnou činnost.

- **TfrmMain:** Formulář hlavního grafického uživatelského rozhraní, samozřejmě obsahuje i obslužný kód tlačítek – načtení .ini souboru (*Load*, viz *TdmKonfig*) a spuštění exportu (*Start*, viz *TdmExportManager*).
- **TdmExportManager:** Zásadní třída, jejímž účelem je kompletní správa procesu exportu a odesílání dat. Vstupním bodem je funkce *Start*, která pomocí *TMultiQuery* provede dotaz do tabulky *LG_EXTZAR_EXPORT* a získá data určená k exportu, označovaná jako *QryExport*. Ostatní třídy pak s těmito daty pracují pomocí ukazatele označovaného *pQryExport*. Zde jsou rovněž vytvořeny textové buffery pro ukládání výsledků (ukazatel *pBuffer*) a informací o chybách (ukazatele *pKodChyby* a *pTextChyby*), do nichž jsou ukládány výsledky zpracování a exportu dat. Následně sekvenčně prochází *QryExport* a nad každou klinickou zprávou zavolá metodu *ZpracujZaznam*, která na základě parametrů exportu, získaných společně s daty z databáze, vybere dle příslušné logiky správný formát exportu a způsob odeslání či uložení a následně využije

příslušných tříd k provedení těchto akcí. Zároveň zajišťuje výpisy a logování informačních a chybových zpráv.

- **TdmKonfig:** Servisní třída pro konfiguraci aplikace po jejím spuštění. Provádí načtení .ini souboru s konfigurací a na jeho základě nastavuje veškeré parametry (funkce *Load*), vytváří a ruší připojení k databázím (*ConnectDB*, *DisconnectDB*, *ConnectDBIMG*, *DisconnectDBIMG*), nastavuje cesty a jména výstupních souborů (*GetFileName*) a pomocí několika dalších metod řeší správné nastavení a použití kódových stránek pro češtinu. Hlavičkový soubor této třídy definuje konstanty (především řetězce reprezentující názvy různých parametrů vytvářených síťových připojení, formátů a způsobů komunikace apod.) pro celou aplikaci.
- **ErroFiles:** Samostatná část, která však není třídou; je includována v každé ze tříd a obsahuje konstanty a metody pro informační a chybové výpisy. V hlavičkovém souboru jsou definovány konstanty pro chybové kódy, v těle pak funkce pro tři úrovně výpisu (*Info*, *Error* a *FatalError*). V závislosti na nastavení probíhá výpis do informačního okna v GUI nebo do souboru.
- **TdmAutoRepeat:** Třída, která je instanciována v případě, že při exportu či odesílání některých dat dojde k chybě. Dojde k zapsání záznamu do logu a vrácení dat, které nebyly úspěšně zpracovány, do tabulky LG_EXTZAR_EXPORT tak, aby při příští dávce došlo k opakovanému pokusu.
- **TdmCertMgr:** Instancuje komponentu TipsCertMgr pro správu certifikátů, čímž zpřístupní úložiště certifikátů a zprostředkovává metody pro přístup k certifikátům.
- **TcertKarta:** Třída obsahující metodu pro ověření kódu PIN u čipové karty s certifikátem pro systém MEX.
- **TfrmZprava:** Pomocný formulář používaný při převodu z plaintextu na formát RTF a vice versa.

9.3.2 Třídy pro zpracování dat

Sada tříd určená pro zpracování dat z databáze, načtených do QryExport. Třída TdmExportManager je spouští v závislosti na zvoleném formátu dat a předává jim přístup k datům a bufferům pomocí parametrů jejich funkcí (viz kapitola 9.3.1). Data jsou z TMultiQuery vhodnou logikou převedena do požadovaného formátu a uložena do bufferu, v případě výskytu chyby je informace rovněž uložena do příslušných bufferů a

řídící třída se postará o jejich zpracování. Vstupním bodem většiny tříd je metoda *Export*, které jsou parametrem předány ukazatele či reference právě na *QryExport* a všechny buffery.

- **TdmDastaExport:** Důležitá třída, která zajišťuje kompletní převod dat z databáze do formátu dle standardu DASTA 3. Obsahuje řadu funkcí, reprezentujících jednotlivé části „business logiky“ standardu DASTA, jejichž pomocí je postupně zkonstruován celý XML dokument odpovídající tomuto standardu a obsahující exportovaná data. Samozřejmě je využita *TICEXMLComponent* komponenta pro práci s XML a několik dalších instancí *TMultiQuery* pro získání dodatečných dat z DB pomocí dalších SQL dotazů. Výsledek je samozřejmě uložen do příslušných bufferů.
- **TdmMikyskaExport:** Tato třída obsahuje logiku pro export jednoho specifického typu klinických událostí (konkrétně Laboratorní vyšetření) do určitého proprietárního formátu, pravděpodobně pro jednu konkrétní laboratoř nebo podobně.
- **TdmRTFExport:** Velice jednoduchá třída pro prosté uložení textu lékařské zprávy z dané klinické události do výstupního bufferu.
- **TdmImageData:** Třída pro práci s obrazovými daty, resp. pro jejich načítání z databáze. Obsahuje funkce pro zpracování obrazových dat uložených v databázi různými způsoby; LOB nebo BLOB, různé druhy komprese atd. Rovněž načítá i příslušná metadata. Neobsahuje funkci *Export*, její funkce jsou používány v ostatních třídách.
- **TdmDastaResult:** Pomocná třída, sloužící ke zpracování XML odpovědi z brány IZIP (viz kapitola 9.3.3, *TdmIZIPSender* a *TdmCheckIzipRC*). Obsahuje několik metod na parsing této odpovědi za účelem dalšího programového zpracování.

9.3.3 Třídy pro komunikaci, odesílání a ukládání dat

Tyto třídy zajišťují komunikaci se vzdálenými servery, na které jsou data odesílána, případně jejich přenos jiným způsobem či ukládání do souborů. Vstupním bodem je zpravidla třída *Process*, která má stejně jako třída *Export* v minulé kapitole parametry v podobě ukazatelů či referencí na *QryExport*, textový buffer na zpracovaná data a textový buffer na chyby. Z *QryExport* se získávají pouze případné doplňující

údaje a parametry odeslání, samotná data jsou již připravena v bufferu jednou z metod popsaných v kapitole 9.3.2. Princip práce s chybami je totožný.

- **TdmCheckIzipRC:** Jedna ze tříd pro komunikaci s elektronickou zdravotnickou knížkou IZIP. Tato třída slouží pro ověření pacienta, jemuž přísluší právě zpracovávaná data, oproti systému, a to na základě jeho rodného čísla. Pomocí funkce *SestavXML* je vytvořen XML dokument s dotazem v příslušném formátu, ten je pomocí *FillIdcZpSezn* a za využití třídy *TIzipSender* odeslán na bránu systému IZIP a následná odpověď je zpracována funkcí *ZpracujOdpoved*. Na základě odpovědi je určen stav konta a registrace pacienta v IZIP (neregistrován, neaktivní účet, aktivní účet apod.) a v závislosti na tom je komunikace buď ukončena, nebo nastaven potřebný stav registrace či konta (funkce *SetStav* a *SetStavRegistrace*).
- **TdmIZIPSender:** Třída *TdmIzipSender* je určena pro odesílání dat na elektronickou zdravotnickou knížku IZIP. Přístupové údaje na bránu IZIP jsou získány z *QryExport*. Samotné připojení na vzdálený server (bránu) a odeslání dat provede třída *TIzipSender*. Po ukončení odesílání vrátí brána XML s výsledkem (potvrzením přijetí dat či důvodem jejich odmítnutí), které třída přijme a pomocí metody *SaveInputOutputFiles* uloží na disk pro případné pozdější ověření či dohledání chyb.
- **TdmMednetSender:** Třída pro odeslání dat do systému MedicalNET. Vytvoří dočasný soubor s daty, načte URL serveru z *QryExport* a za použití knihovny *MedicalNET.lib*, která je součástí projektu a obsahuje funkce pro komunikaci s MedicalNET servery, odešle data na daný server. Při zdárném průběhu samozřejmě následně smaže dočasný soubor.
- **TdmMEXSender:** Klientská třída pro komunikaci s webovou službou systému MEX první generace (viz kapitola 7.2) za použití protokolů SOAP a HTTPS. Prvním krokem je inicializace připojení přes oba protokoly pomocí funkcí *InitSOAPS* a *InitHTTPS*, které nakonfigurují parametry připojení (adresy, porty, čipová karta s certifikátem, přihlašovací údaje...). Následně je provedeno přihlášení na server (funkce *Login*) a odeslání dat pomocí funkce *SendTransfer*, v případě obrazových dat i *FileTransfer*. Po ukončení proběhne logout pomocí stejnojmenné funkce. Třída dále obsahuje určité podpůrné funkce, průběh komunikace je samozřejmě důsledně logován.

- **TdmRadiomedSender:** Klientská třída odesílající data ve formátu DASTA firmě Radiomed přes webovou službu pomocí protokolu SOAP. Používá komponentu TipsSOAPS (viz kapitola 9.2.1) ke konfiguraci připojení přes tento protokol (funkce *InitSOAPS*, jež nastaví parametry pro průchod firewallem apod.) a následně i odeslání dat z bufferu (funkce *SendData*, která sestrojí SOAP hlavičku, připojí zprávu a odešle na interface). Pokud je nastaveno logování, jsou odeslaná i přijatá data uložena pro případnou kontrolu či odladění chyb.
- **TdmSendMail:** Data jsou získána z bufferu, potřebné parametry pro odeslání z QryExport (např. adresa) a konfigurace aplikace (např. přístup k SMTP serveru). Metoda *Process* zkonstruuje hlavičku mailu, pomocí funkce *ConstructText* a dalších pomocných funkcí pak i tělo zprávy, zašifruje zprávu pomocí klíče získaného z úložiště certifikátů (viz kapitola 9.3.1, *TdmCertMgr*) a odešle na adresu specifikovanou v parametrech exportu.
- **TdmSendRequestToModality:** Třída pro odeslání obrazových dat na server DICOM. Odeslání je řešeno pomocí kódu v jazyce PL/SQL přímo v databázi, tato třída obsahuje pouze metodu pro jeho spuštění a použití za pomoci komponenty *TMultiPackage*, která umožňuje přístup k balíčku PL/SQL, obsahujícímu danou funkcionalitu.
- **TdmSaveToFile:** Ukládá textová data z bufferu do souboru, název souboru je určen na základě *TdmKonfig* (viz kapitola 9.3.1).
- **TdmSaveToFileImg:** Ukládá obrazová data do souboru. Po spuštění *Process* vytvoří instanci *TdmImageData* (viz kapitola 9.3.2), pomocí jejích metod načte požadovaná obrazová data z databáze a následně je exportuje do souboru.
- **TizipSender:** Tato třída zprostředkovává komunikaci se vzdáleným systémem elektronické zdravotnické knížky IZIP. Data k odeslání a potřebné parametry (URL a přihlašovací údaje) získá při instanciaci přes parametry konstrukturu a následně provede odeslání pomocí metody *Send*, veškerá další logika je obsažena ve volajících třídách (viz *TdmCheckIzipRC* a *TdmIZIPSender*). K tomuto účelu využívá externí knihovnu *izipc.lib*, obsaženou v projektu, která poskytuje funkce pro samotnou komunikaci se serverem (bránou) systému IZIP.

10. Realizace rozšíření

Jak bylo zmíněno v kapitole 8, primárním cílem bylo rozšířit aplikaci AutoExport o možnost exportu libovolných dat z databáze do vhodného formátu, který bude možné předat k dalšímu využití či zpracování. Sekundárním cílem pak je vyřešit právě způsob přenosu dat exportovaných do tohoto formátu ze serveru systému Medicalc na ZČU a případně do dalších institucí.

Požadavky je možné definovat následovně:

- Absolutní nezávislost na datech a použitelnost pro export dat ze skutečně libovolného dotazu provedeného nad databází systému.
- Textový výstup, formát vhodný pro síťový přenos i uložení do souboru.
- V první fázi export do souboru, výhledově implementace vhodného síťového přenosu.
- Co nejlepší integrace do původního řešení, zároveň však co nejmenší úpravy stávajícího kódu, samozřejmě nenarušení původní funkčnosti.

10.1 Výstupní formát

Jako nejvhodnější výstupní formát byl zvolen standard XML z důvodu vysoké adaptability, strojové i lidské srozumitelnosti a širokých možností přenosu. Původní úvahou bylo od začátku použít stejný formát, ve kterém budou data dodávána do SITS (viz kapitola 8), nicméně toto bylo zavrženo jednak z toho důvodu, že spolupráce se SITS ještě není dořešena a formát tak nebyl znám a jednak proto, že aby byl formát použitelný pro libovolná data, nelze definovat názvy XML elementů předem; musí být určeny na základě toho, jaká data jsou exportována. Proto byl navržen generický formát, který je v důsledku velmi jednoduchý, nicméně funkční, univerzální a takto byl schválen pro použití:


```

<?xml version="1.0" encoding="WINDOWS-1250"?>
- <data>
  - <row>
    <ezexp_id>77</ezexp_id>
    <prijmeni>TESTOVACÍ</prijmeni>
    <jmeno>PACIENT</jmeno>
    <pacient_id>2</pacient_id>
    <narozen>1901-01-01</narozen>
    <pohlavi>M</pohlavi>
    <statni_prisl>ČR</statni_prisl>
    <identif_pac>010101011</identif_pac>
    <poj_datum_od>1992-01-01</poj_datum_od>
    <poj_datum_do>3000-01-01</poj_datum_do>
    <kod_platce>999</kod_platce>
    <pojistovna>999 Samoplátce</pojistovna>
    <nazev_spu>Návštěva ambulance / ordinace</nazev_spu>
    <skupina_tpu>M</skupina_tpu>
    <url>C:\Medicalc\Autoexport\Data\</url>
    <adresa1>Plzeňská 15</adresa1>
    <posta>Plzeň</posta>
    <psc>30100</psc>
    <telefon>377 001 001</telefon>
    <extzar_nazev>File</extzar_nazev>
    <extzar_id>2</extzar_id>
    <komunikace>FILE</komunikace>
    <export_format>GENERIC</export_format>
    <globkod>File</globkod>
    <klinudal_id>38</klinudal_id>
    <poz_intext>N</poz_intext>
    <poz_prac_id>0</poz_prac_id>
    <kod_typu>001</kod_typu>
    <kod_spec>000</kod_spec>
    <xplatce_id>3</xplatce_id>
    <datum_prov>2015-04-16T11:17:00</datum_prov>
    <stav_provedeni>P</stav_provedeni>
    <stav_dokumentace>D</stav_dokumentace>
    <prov_klinodd_id>1</prov_klinodd_id>
    <prov_pracoviste>Ambulance</prov_pracoviste>
    <prov_odb>001</prov_odb>
    <prov_druh>A</prov_druh>
    <prov_cislo>01</prov_cislo>
    <prov_icp>00000001</prov_icp>
    <poz_klinodd_id>0</poz_klinodd_id>
    <prov_lek_prijmeni>Správce KIS</prov_lek_prijmeni>
    <prov_lek_jmeno>Správce</prov_lek_jmeno>
    <akt_datum_cas>2015-06-23T20:16:23</akt_datum_cas>
    <stav>P</stav>
    <datum_exp>2015-06-23T20:16:23</datum_exp>
    <datum_urc>2015-06-23T20:15:41</datum_urc>
    <ezexp_id_1>77</ezexp_id_1>
    <zpracovavat>A</zpracovavat>
    <poz_extzar_id>0</poz_extzar_id>
    <prov_extzar_id>0</prov_extzar_id>
    <akt_login_username>KIS</akt_login_username>
    <extzar_pracov_id>0</extzar_pracov_id>
    <typ_exportu>T</typ_exportu>
  </row>
</data>

```

Obr. 10.1 – Navržený generický výstupní formát

Element *Data* je kořenový, pro každý exportovaný řádek je vytvořen element *row* a v něm jsou obsaženy elementy, jejichž názvy odpovídají názvům sloupců entity, která je výsledkem dotazu a obsažená data samozřejmě datům na daném řádku. Formát čísel je na žádost zadavatele český, tedy s desetinnou čárkou a oddělovačem tisíců

v podobě mezery, formát data a času je rovněž na žádost zadavatele specifický; víceméně odpovídá normě ISO, avšak není uvedeno časové pásmo.

Prázdné sloupce jsou vynechány, toto chování je samozřejmě možné v případě potřeby nastavit. Některé sloupce v příkladu obsahují interní parametry exportu, které by pravděpodobně bylo ideální rovněž vynechat, nicméně tím by byl porušen požadavek absolutní generičnosti exportu.

10.2 Implementace rozšíření

Rozšíření bylo navrženo tak, aby architekturou odpovídalo původní aplikaci a plynule navazovalo na dosavadní funkčnost. Z toho plyne jedna nová třída pro zpracování dat (viz kapitola 9.3.2), doplnění příslušných řídicích tříd (viz kapitola 9.3.1) a výhledově i další třídy pro komunikaci a odesílání dat (viz kapitola 9.3.3 a kapitola 11).

10.2.1 Třída TdmDataSetExport

Nová třída pro zpracování dat, určená pro tvorbu navrženého generického formátu. Dědí z TDataModule, což umožňuje pohodlnou práci s potřebnými komponenty, nicméně současně době využívá pouze ICE XML pro konstrukci XML dokumentu.

Vstupním bodem je metoda *Export* (*TMultiQuery *pQryExport, AnsiString &pBuffer, AnsiString &pKodChyby, AnsiString &pTextChyby*), což odpovídá standardu aplikace.

- Data určená k exportování jsou získána pomocí instance třídy TMultiQuery, kterou vytvoří řídicí třída, jež zároveň spustí SQL dotaz nastavený v této instanci a rovnou ji tak naplní daty. Do funkce je předán ukazatel na tuto instanci a je tedy možné z ní rovnou získávat data ke zpracování.
- Reference na buffer zpřístupňuje textový buffer, rovněž vytvořený volající třídou. Do něj je ukládán výstup v podobě textu zformátovaného dle požadavků.
- Přes referenci na řetězec s kódem chyby je možné předat volající funkci jednoznačný identifikátor vzniklé chyby. Ty jsou definovány v ErrorFiles (viz kapitola 9.3.1) jakožto konstanty (viz níže).
- Přes referenci na řetězec s textem chyby je v případě chyby volající funkci předán kompletní textový popis výjimky, která byla vyhozena.

Po vytvoření kořenového elementu následuje sekvenční postup, kdy metoda prochází postupně data z TMultiQuery. Přístup k datům v TMultiQuery je pomocí ukazatele na objekt typu TFields, odpovídající jednomu řádku výsledku dotazu, jenž je de facto seznamem ukazatelů na objekty typu TField, což je reprezentace jednoho pole výsledku. Ten obsahuje nejen data, ale i název a datový typ pole, na jeho základě je tedy vytvořen příslušný XML element a zařazen do hierarchie dokumentu.

Data jsou naformátována pomocí za tímto účelem vytvořených metod, zatím je dle požadavků zadavatele implementováno formátování Float a DateTime:

- Float je formátován pomocí funkce *processFloat (TField *pField)*, jež přes ukazatel načte z objektu TField původní číselnou hodnotu a pomocí funkce *FormatFloat* z ní vytvoří řetězec s požadovaným formátem (desetinná čárka a oddělovače tisíců) a vrátí.
- DateTime je formátován pomocí funkce *processDateTime (TField *pField)*, jejíž princip je obdobný; na základě toho, jestli původní pole obsahuje pouze datum, nebo i čas, vytvoří řetězec s požadovaným formátem a vrátí.

Ostatní data jsou převzata jakožto řetězec, nicméně v případě potřeby je samozřejmě možné doplnit formátování kteréhokoli dalšího typu.

Po dokončení je zkonstruovaný XML dokument převeden na řetězec a uložen do bufferu pro další zpracování volající metodou. Pokud při zpracování dojde k výjimce, jsou informace samozřejmě rovněž uloženy do příslušných bufferů a informace vypsána do stavového okna či logována pomocí standardních metod ze souboru ErrorFiles.

10.2.2 Úpravy stávajícího kódu

Aplikaci bylo samozřejmě nutné přizpůsobit tak, aby nové rozšíření podporovala a proto byly provedeny následující úpravy:

- Do hlavičkového souboru třídy **TdmKonfig** byly doplněny konstanty pro export do tohoto formátu, nazvaného „GENERIC“ či „FORMAT_GENERIC“. Rovněž byla upravena metoda pro přidělování jmen souborů *GetFileName*, která přidělovala názvy pouze dle standardu DASTA, což bylo pro potřeby tohoto rozšíření nevhodné. Nyní při použití formátu GENERIC vytváří jména souborů kombinací prefixu „GEN“ a unixové časové značky. Pokud je vytvořeno více

souborů v jedné sekundě, přidá navíc pořadovou číslici, aby nedošlo k přepsání. Název souboru v novém formátu je tedy např. „GEN1435083384.XML“, přičemž je uložen do adresáře, nakonfigurovaného v nastavení aplikace, stejně jako soubory formátu DASTA.

- Hlavičkový soubor **ErrorFiles** byl rozšířen o chybovou konstantu nového rozšíření. Konstanty mají podobu např. „GEN-00000“, kdy prefix identifikuje místo, kde se chyba stala (v tomto případě export dat do generického formátu) a číslo konkrétní chyby (v tomto případě blíže nespecifikovaná výjimka). Tyto chybové konstanty se ukládají do bufferu KodChyby (viz výše) a zpracovává je volající metoda.
- Řídící třída **TdmExportManager** byla doplněna tak, aby uměla s novým formátem exportu pracovat. Je-li tedy v datech k exportu, získaných z tabulky LG_EXTZAR_EXPORT pomocí TMultiQuery QryExport, požadován export do formátu GENERIC, výstup je zformátován pomocí nově vytvořené třídy TdmDataSetExport. V závislosti na zvoleném typu komunikace je pak exportován do souboru (viz výše) či odeslán e-mailem. Odesílat nově vytvořený formát na servery různých služeb popsanych výše samozřejmě nelze.

10.3 Shrnutí současné funkčnosti

Primární požadavek je v tuto chvíli splněn a upravená verze aplikace byla předána firmě Medicalc software s.r.o. Aplikace je nakonfigurována tak, aby byla schopna exportovat klinické události ze systému WinMedicalc v nově navrženém formátu a ukládat je do souborů pro fyzický transport, nebo zasílat pomocí e-mailu. Za současného stavu se počítá především s dávkovým fyzickým transportem dat ve formě souborů.

Novou funkčnost je možno dle případných požadavků upravit či doplnit a především použít jiným způsobem; bude-li třeba, vstupem může být nejen klinická událost, ale výsledek jakéhokoli SQL dotazu nad databází provedeného pomocí komponenty TMultiQuery.

11. Další plán rozšíření

Druhou, zatím neimplementovanou částí rozšíření je vytvoření síťového komunikačního kanálu pro data v novém generickém formátu. Samozřejmě by bylo možné využít již funkčního e-mailového odesílání a vytvořit nějaký automatický parser pro zpracování přijatých zpráv, nicméně takovéto řešení nelze považovat za elegantní ani praktické pro použití.

Jako řešení se samozřejmě nabízí použití webové služby, se kterou bude komunikováno pomocí protokolu SOAP přes HTTP/HTTPS, což je využíváno pro některé již existující komunikační kanály a jedná se o ideální způsob přenosu dat ve formátu XML. Na straně aplikace AutoExport půjde o další klientskou třídu pro komunikaci a odesílání dat (viz kapitola 9.3.3), která naváže jednosměrnou komunikaci s webovou službou na straně ZČU či jiného případného zařízení, určenou pro příjem a ukládání či zpracování dat a odešle na ni data. Implementace této webové služby může být díky vlastnostem protokolu SOAP víceméně libovolná, dokud bude poskytovat příslušné rozhraní webové služby.

11.1 Návrh webové služby

Ačkoli je celá tato část práce zatím ve fázi návrhu, parametry, požadavky na funkčnost i předpokládané technické řešení webové služby pro příjem dat jsou již z většiny známy a lze tak předeslat podobu řešení.

11.1.1 Požadavky na funkčnost webové služby

Webová služba bude určena čistě pro příjem dat, bude tedy podporovat pouze jednosměrnou komunikaci, kdy na ni klientská třída AutoExport odešle data a nebude vracet žádnou odpověď. Server s webovou službou by měl být následně schopen data zpracovat některými z následujících způsobů:

- Rozparsovat XML a uložit do databáze včetně dalších metadat (časová značka přijetí, odesílající klient apod.) Tato možnost předpokládá, že budou zasílána data o známé struktuře (např. klinické zprávy), pouze převedená do nově navrženého formátu.
- Uložit kompletní XML do databáze jakožto řetězec, opět včetně příslušných metadat.

- Přeposlat data jinému systému (např. SITS), samozřejmě v případě potřeby před přeposláním konvertovat na formát požadovaný koncovým systémem.

V případě implementace více možností by volba způsobu zpracování měla proběhnout automaticky na základě obsahu přijatých dat dle pravidel, předem definovaných v aplikaci.

Vzhledem k citlivosti lékařských dat je nutné zajistit bezpečnost, což znamená autorizaci, autentikaci a šifrovanou komunikaci. Všechny tyto funkce zajišťuje rozšíření protokolu SOAP s názvem Web Service Security, které již bylo zmíněno výše.

11.1.2 Technické řešení webové služby

Z technického hlediska bude server webové služby představován pravděpodobně webovou aplikací, postavenou na platformě Java Enterprise Edition, jako databáze bude sloužit Oracle. Samozřejmě by bylo možné využít i jiných technologií (např. PHP + MySQL), pokud by to bylo z nějakého důvodu požadováno.

Aplikace bude nasazena na fyzickém či virtuálním serveru na ZČU a bude poskytovat příslušné rozhraní, definované standardně pomocí WSDL. Rovněž bude zajišťovat veškeré funkce pro zpracování a ukládání dat.

11.1.3 Rozhraní webové služby

Předběžná podoba rozhraní uvažované webové služby je definována pomocí WSDL na obrázku 11.1. Obrázek slouží pouze pro představu, nejsou tedy ještě vyřešeny některé dílčí požadavky, jako je zmiňované zabezpečení.

```
<?xml version="1.0"?>
- <definitions targetNamespace="cz.zcu.wSDL" name="Accepts and processes data in generic format from WinMedicalc."
  xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsd1="cz.zcu.xsd"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/"
  xmlns:tns="cz.zcu.wSDL">
  <!-- definition of datatypes -->
  - <types>
    - <schema targetNamespace="cz.zcu.xsd" xmlns="http://www.w3.org/2000/10/XMLSchema">
      - <element name="data">
        - <complexType>
          - <all>
            <element name="value" type="string"/>
          </all>
        </complexType>
      </element>
      - <element name="resultCode">
        - <complexType>
          - <all>
            <element name="value" type="int"/>
          </all>
        </complexType>
      </element>
    </schema>
  </types>
  <!-- request messages -->
  - <message name="receiveGenericXML">
    <part name="data" type="xsd:data"/>
  </message>
  <!-- server's services -->
  - <portType name="medicalcExport">
    - <operation name="receiveGenericXML">
      <input message="tns:receiveGenericXML"/>
    </operation>
  </portType>
  <!-- server encoding -->
  - <binding name="medicalcExport_webservices" type="tns:medicalcExport">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/https" style="rpc"/>
    - <operation name="receiveGenericXML">
      <soap:operation soapAction="urn:xmethods-delayed-quotes#receiveGenericXML"/>
      - <input>
        <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:xmethods-delayed-quotes" use="encoded"/>
      </input>
      - <output>
        <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:xmethods-delayed-quotes" use="encoded"/>
      </output>
    </operation>
  </binding>
  <!-- access to service provider -->
  - <service name="zcu">
    - <port name="zcu_0" binding="medicalcExport_webservices">
      <soap:address location="https://147.228.254.252/medicalcExport"/>
    </port>
  </service>
</definitions>
```

Obr. 11.1 – Návrh rozhraní webové služby

11.2 Návrh dalšího rozšíření *AutoExport*

Rozšíření bude spočívat v doplnění další třídy a úpravách stávajícího kódu obdobně, jako tomu bylo při vytváření třídy pro generování nově navrženého formátu.

11.2.1 Třída *TdmZCUSender*

Tato třída bude kopírovat principy, nastavené již existujícími třídami pro komunikaci pomocí webových služeb. Vstupním bodem bude funkce *Process* (*TMultiQuery *pQryExport, AnsiString &pBuffer, AnsiString &pKodChyby, AnsiString &pTextChyby*), jež načte zformátovaná data z bufferu a zavolá funkci *SendData*.

Funkce *SendData* využije komponenty *TipsSOAPS* k sestavení SOAP požadavku a jeho odeslání na server webové služby. Sestavení hlavičky, obálky i těla požadavku, stejně jako jeho samotné odeslání, je v této komponentě prováděno pomocí příslušných funkcí a je poměrně intuitivní.

11.2.2 Úpravy stávajícího kódu

Samozřejmě bude nutné doplnit třídu *TdmKonfig* a *ErrorFiles* o konstanty pro komunikaci typu „ZCU“ a její chybové stavy. Zásadní pak bude opět rozšíření *TdmExportManager* o podporu nového způsobu komunikace.

12. Závěr

Práce dokonale splnila svůj primární účel. Poskytuje náhled do problematiky zdravotnických dat se zaměřením na standardy pro ukládání a výměnu těchto dat, a to včetně popisu těch standardů, které jsou relevantní pro prostředí českého zdravotnictví a v něm používaných zdravotnických informačních systémů. Analyzuje různé verze informačního systému Medicalc, přičemž se zaměřuje především na způsoby a možnosti exportu, importu a obecně výměny dat při použití tohoto systému.

V druhé části se pak práce zabývá možnostmi rozšíření této funkčnosti o nový formát exportu dat, který bude možné využít pro přenos některých dat z tohoto systému na ZČU a některé jeho části zároveň budou užitečné i pro dodavatele systému, tedy společnost Medicalc software s.r.o. Diskutuje navržené možnosti a jejich výhody a nevýhody a u některých z nich ověřuje možnosti jejich realizace.

U vybrané možnosti v podobě úpravy aplikace AutoExport provádí analýzu současného stavu této aplikace včetně zběžné dokumentace a zabývá se rozšířením její funkčnosti. Navrhuje nový formát, vhodný pro export libovolných dat ze systému a implementuje funkčnost pro generování a ukládání tohoto formátu, kterážto je nyní plně k dispozici. Na závěr se zabývá možnostmi přenosu tohoto formátu po síti, které však zatím nebyly zrealizovány.

Použité zdroje

- [1] DAVENPORT, Thomas H.; PRUSAK, Laurence. Working knowledge : how organizations manage what they know. Boston : Harvard Business School, 1998. xxiv, 201 s. 0-87584655-6.
- [2] VYMĚTAL, Jan; DIAČIKOVÁ, Anna; VÁCHOVÁ, Miriam. Informační a znalostní management v praxi. Praha : LexisNexis CZ, 2005. 399 s. ISBN 80-86920-01-1.
- [3] KUČEROVÁ, Helena. Definice informace. Data - informace - znalosti. Vyšší odborná škola informačních služeb: Helena Kučerová – domovská stránka [online]. [cit. 2015-04-17]. Dostupné z: <http://web.sks.cz/users/ku/ZIZ/inform1.htm>
- [4] GOTTSCHALK, Petter. Knowledge management. In Jennex, Murray E. (ed.). Knowledge management : concepts, methodologies, tools, and applications [online]. Hershey : Information Science Reference, 2008, s. 130-143.
- [5] Informační systémy ve zdravotnictví. Jihočeská univerzita v Českých Budějovicích, Zdravotně sociální fakulta, Katedra radiologie, toxikologie a ochrany obyvatelstva [online]. [cit. 2015-06-18]. Dostupné z: http://www.zsf.jcu.cz/cs/katedra/katedra-radiologie-toxikologie-a-ochrany-obyvatelstva/informace-katedry/informace-pro-studenty/ucebni_texty/ochrana-obyvatelstva-se-zamerenim-na-cbrne-aplikovana-radiobiologie-a-toxikologie-krizova-radiobiologie-a-toxikologie/informacni-systemy-ve-zdravotnictvi/view
- [6] KRÁL, Jaroslav a Michal ŽEMLIČKA. Kvalita dat a informací – základní omezení IT ve veřejné správě. Risk-Management.cz [online]. [cit. 2015-04-18]. Dostupné z: <http://www.risk-management.cz/clanky/kral-zemlicka-kvalita-dat-a-informaci-zakladni-omezeni-IT-ve-verejne-sprave.pdf>
- [7] WANG, Richard Y, Mostapha ZIAD a Yang W LEE. Data quality. Boston, Mass.: Kluwer Academic, 2002, xv, 167 s. Kluwer international series on advances in database systems, 23.
- [8] PIPINO, Leo L., Yang W. LEE a Richard Y. WANG. Data Quality Assessment. COMMUNICATIONS OF THE ACM. 2002, (45): 211-218. Dostupné také z: <http://web.mit.edu/tdqm/www/tdqmpub/PipinoLeeWangCACMApr02.pdf><http://web.mit.edu/tdqm/www/tdqmpub/PipinoLeeWangCACMApr02.pdf>

- [9] Data Security. Techopedia [online]. 2015 [cit. 2015-04-24]. Dostupné z: <http://www.techopedia.com/definition/26464/data-security>
- [10] Definice Interoperability [online]. 2015 [cit. 2015-04-24]. Dostupné z: <http://interoperability-definition.info/cs/>
- [11] Health Information Technology Standards [online]. 2015 [cit. 2015-05-04]. Dostupné z: http://www.phdsc.org/standards/health-information/D_Standards.asp
- [12] ASPDEN, Philip. Patient safety: achieving a new standard for care. Washington, D.C.: National Academies Press, 2004, xx, 528 s.
- [13] DASTA [online]. 2012 [cit. 2015-05-06]. Dostupné z: www.dastacr.cz
- [14] NAIDR, Jan P. Datový standard MZ ČR a NČLP. Katedra managementu Fakulty managementu VŠE v Praze [online]. [cit. 2015-05-06]. Dostupné z: <http://www.fm.vse.cz/km/wp-content/uploads/2009/10/VEH598-06-datovystandard.pdf>
- [15] SEDLÁČKOVÁ, Eva. Datový standard zdravotnických informačních systémů. Brno, 2012. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=52456.
Bakalářská práce. Vysoké učení technické v Brně. Vedoucí práce Ing. Petr Fedra.
- [16] Datový standard MZ ČR. NÁHLÍK, Jan. Gerstner laboratory [online]. 2006 [cit. 2015-05-07]. Dostupné z: http://gerstner.felk.cvut.cz/biolab/33LI/Slides/Datovy_standard.pdf
- [17] DICOM [online]. 2015 [cit. 2015-05-15]. Dostupné z: <http://medical.nema.org/>
- [18] Health Level Seven International [online]. 2015 [cit. 2015-05-15]. Dostupné z: <http://www.hl7.org/>
- [19] HEITMANN, Kai U., Dr. Bernard BLOBEL a Prof. Dr. J. DUDECK. HL7 Komunikační standard ve zdravotnictví: Krátký úvod a informace [online]. 1999 [cit. 2015-05-15]. Dostupné z: http://www.hl7cr.eu/file/13/HL7_komunikace.pdf
- [20] Health Level Seven (HL7). Public Health Data Standards Consortium [online]. 2015 [cit. 2015-05-15]. Dostupné z: <http://www.phdsc.org/standards/health-information/hl7.asp>
- [21] Architecture of SNOMED: Its Contribution to Medical Language Processing. Proc Annu Symp Comput Appl Med Care. 1986, (Oct 26): 74-80. Dostupné z: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2245000/pdf/procascamc00020-0083.pdf>
- [22] SNOMED CT and HL7: Bringing Standards Together. Health Level Seven International [online]. 2012 [cit. 2015-05-25]. Dostupné z:

http://www.hl7.org/documentcenter/public_temp_86F894C2-1C23-BA17-0C69948995F71502/calendarofevents/himss/2012/SNOMED%20CT%20and%20HL7%20--%20Bringing%20Standards%20Together.pdf

[23] Evaluation of LOINC as a reference terminology for clinical document types: A case report of an outpatient EHR. Academia.edu [online]. 2015 [cit. 2015-05-25].

Dostupné z:

https://www.academia.edu/2056669/Suitability_of_LOINC_Document_Ontology_as_a_reference_terminology_for_clinical_document_types_A_case_report_of_a_research-oriented_EHR

[24] Suitability of LOINC Document Ontology as a reference terminology for clinical document types: A case report of a research-oriented EHR. Academia.edu [online].

2015 [cit. 2015-05-25]. Dostupné z:

https://www.academia.edu/2056669/Suitability_of_LOINC_Document_Ontology_as_a_reference_terminology_for_clinical_document_types_A_case_report_of_a_research-oriented_EHR

[25] About LOINC. LOINC from Regenstrief [online]. 2015 [cit. 2015-05-25].

Dostupné z: <https://loinc.org/background>

[26] BUI, Alex A a Ricky K TAIRA. Medical imaging informatics. New York: Springer, 2010, xxii, 446 p. ISBN 1441903852-.

[27] UMLS Reference Manual [online]. 2009 [cit. 2015-06-17]. Dostupné z:

<http://www.ncbi.nlm.nih.gov/books/NBK9676/>

[28] Medicalc4 - nemocniční informační systém. Medicalc [online]. 2015 [cit. 2015-06-17]. Dostupné z: <http://www.medicalc.cz/produkty/medicalc4-52>