

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Aleš Brezovar

ANALIZA HL7 STANDARDA ZA IZMENJAVO PODATKOV V BOLNIŠNIČNIH
INFORMACIJSKIH SISTEMIH

DIPLOMSKO DELO NA
VISOKOŠOLSLEM STROKOVNEM ŠTUDIJU

Mentor: izr. prof. dr. Miha Mraz

Somentor: pred. dr. Jelena Ficzkó

Ljubljana, november 2010



Št. naloge: 00020/2010

Datum: 01.10.2010

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **ALEŠ BREZOVAR**

Naslov: **ANALIZA HL7 STANDARDA ZA IZMENJAVO PODATKOV V
BOLNIŠNIČNIH INFORMACIJSKIH SISTEMIH
ANALYSIS OF HL7 STANDARD USED FOR MEDICAL DATA
EXCHANGE**

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Kandidat naj v svojem delu analizira HL7 standard za izmenjavo podatkov v bolnišničnih informacijskih sistemih. Pri tem naj se kandidat usmeri predvsem na razpoložljive jezike, ki jih HL7 standard ponuja in oceni možnosti njihove uporabe.

Mentor:

prof. dr. Miha Mrz

Dekan:

prof. dr. Nikolaj Zimic

Somentor:

pred. dr. Jelena Ficzkó



IZJAVA O AVTORSTVU
diplomskega dela

Spodaj podpisani/-a ALEŠ BREZOVAR,

z vpisno številko 63050460,

sem avtor/-ica diplomskega dela z naslovom:

ANALIZA HL7 STANDARDA ZA IZMENJAVO PODATKOV V
BOLNIŠNIČNIH INFORMACIJSKIH SISTEMIH

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal/-a samostojno pod mentorstvom (naziv, ime in priimek)

izr. prof. dr. Mihe Mraz

in somentorstvom (naziv, ime in priimek)

pred. dr. Jelene Ficzo

- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne _____ Podpis avtorja/-ice: _____

Zahvala

Ob pripravi diplomskega dela bi se rad zahvalil svoji družini, ki mi je omogočila študij in mi nudila vsestransko pomoč, da sem delo dokončal.

Posebna zahvala gre mojemu mentorju izr. prof. dr. Mihi Mrazu s Fakultete za računalništvo in informatiko ter somentorici pred. dr. Jeleni Ficzkó z Zdravstvene fakultete Univerze v Ljubljani. Obema se zahvaljujem za pomoč in usmerjanje pri pisanju diplomske naloge.

Kazalo

Povzetek	1
Abstract.....	2
1. Uvod	3
2. Projekt eZdravje	5
2.1 eZdravje – gonilo informatizacije slovenskega zdravstvenega sistema	5
2.2 Obseg projekta eZdravje.....	6
2.3 Pričakovane koristi projekta eZdravje.....	7
2.4 Delovanje elektronskega zdravstvenega informacijskega sistema (eZIS)	8
2.5 Podporni standardi v projektu eZdravju	9
3. Organizacija Health Level Seven	10
3.1 Predstavitev organizacije Health Level Seven	10
3.2 Dosežki organizacije HL7 in njeno delovanje.....	11
3.3 Pregled standardov HL7	12
3.3.1 Standard sporočanja HL7 verzija 2.x	12
3.3.2 Standard sporočanja HL7 verzija 3.x	13
3.3.3 Standard CCOW	13
3.3.4 Standard CDA	14
4. Standardi HL7 za izmenjavo podatkov v zdravstvu	15
4.1 Standardni povpraševalni jezik GELLO.....	15
4.1.1 Uvod	15
4.1.2 Predstavitev jezika Gello.....	16
4.1.2.1 Komentiranje kode	18
4.1.3 Podatkovni tipi jezika Gello	18
4.1.3.1 Enostavni podatkovni tipi.....	18
4.1.3.2 Zbirke	18
4.1.3.3 Tip Tuple	19
4.1.4 Operatorji.....	20
4.1.4.1 Preprosti operatorji	20
4.1.4.2 Operatorji razredov.....	21
4.1.4.3 Operatorji zbirk.....	21
4.1.5 Razredi.....	22
4.2 Standardni jezik Arden Syntax	23
4.2.1 Uvod	23
4.2.2 MLM moduli	24
4.2.2.1 Struktura modulov MLM	26
4.2.2.2 Vzdrževalni del modula.....	26
4.2.2.3 Knjižnični del modula	27
4.2.2.4 Logični del modula.....	28
4.2.3 Podatkovni tipi v »Arden Syntax« jeziku.....	29
4.2.4 Operatorji »Arden Syntax« jeziku.....	30
4.2.4 Logični del modula.....	31
4.2.4.1 Spremenljivke.....	31
4.2.4.2 Pogojni stavek	31
4.2.4.2 Klicanje drugih modulov	32

4.2.4.3 Zaključna izjava.....	32
4.2.5 Procesni del modula.....	33
4.2.5.1 Proces pisanja	33
4.2.5.2 Proces klicanja.....	33
4.2.5.3 Proces vračanja vrednosti	33
4.2.6 Modul za preverjanje alergije na penicilin s komentarjem.....	34
5. Zaključek	35
6. Viri.....	36

Slike

Slika 1 : Celovit pogled na projekt eZdravje [3].	7
Slika 2 : Prikaz komunikacije med vozlišči v zdravstvenem informacijskem sistemu.	9
Slika 3 : Prikaz priporočil HL7.....	10
Slika 4 : Prikaz nivojev v OSI modelu [6].	11
Slika 5 : Prikaz delovanja jezika Gello.....	16
Slika 6 : Podatkovni model v Gellu [13].	23

Seznam uporabljenih kratic in simbolov

ANSI	American National Standards Institute – zasebna neprofitna organizacija, ki nadzoruje razvoj standardov za proizvode, storitve, postopke, sisteme in osebje v Združenih državah Amerike in drugod po svetu.
DICOM	Digital imaging and communications in medicine – standard za upravljanje, shranjevanje, tiskanje in prenos podatkov pri medicinskih oslikavah.
EKG	Elektrokardiogram – grafični zapis električne aktivnosti srca. Kljub novejšim metodam velja EKG še vedno za najbolj razširjeno metodo za odkrivanje različnih obolenj srca.
HL7	Health level seven - prostovoljna, neprofitna organizacija, ki sodeluje pri razvoju mednarodnih standardov v zdravstvu.
ISO	International Organization for Standardization – mednarodna organizacija za standardizacijo.
LAN	Local area network – lokalno računalniško omrežje, ki pokriva manjše območje.
OSI	Open System Interconnection – referenčni model, v katerem je standardizirano medsebojno povezovanje odprtih sistemov in ima definirane funkcije povezovanja v sedmih hierarhičnih slojih.
PACS	Picture archiving and communication system – sistem za arhiviranje in prenos digitalnih slik.
SQL	Structured Query Language – najbolj razširjen in standardiziran povpraševalni jezik za delo s podatkovnim bazami.
WAN	Wide area network – prostrano omrežje, ki pokriva večje geografsko področje.
XML	Extensible Markup Language – jezik za opis strukturiranih dokumentov in podatkov, ki predstavlja tehnološko osnovo za izmenjavo podatkov, objavljanje, in elektronsko poslovanje.

Povzetek

V diplomski nalogi so predstavljeni segmenti projekta eZdravje, ki je v Sloveniji sicer še v fazi razvoja, vendar obeta ob svoji polni uveljavitvi velike spremembe na področju informatizacije zdravstva.

Opisani so cilji projekta eZdravje ter predvidene izboljšave na področju informatizacije zdravstva. Predstavljena je neprofitna standardizacijska organizacija HL7, ki postavlja standarde za poenostavljanje dela s podatki v zdravstvu. Ti standardi so podrobneje opisani, obrazloženo pa je tudi njihovo delovanje.

Podrobneje sem opisal HL7 standarda za prikaz podatkov v zdravstvu. Prvi standard je standardni povpraševalni jezik Gello, ki je namenjen delu s podatkovnimi bazami. Drugi standard pa standardiziran programski jezik Arden Syntax, ki se uporablja v modulih za odločanje, širjenje informacij ter alarmiranje zdravstvenega osebja ob določenih situacijah (MLM moduli).

Ključne besede: projekt eZdravje, organizacija standardov HL7, jezik Gello, jezik Arden Syntax, MLM moduli

Abstract

Segments of the project eZdravje (eHealth) are presented in the thesis. The project is still in the phase of development in Slovenia, however it raises expectations of substantial changes in the field of the computerisation of health care once it is fully implemented.

The goals of the project eZdravje are described as well as the foreseen improvements in the field of computerisation of health care. A non-profit standardisation organisation HL7, which is setting the standards for simplification of the processing of data in health care, is presented. Those standards have been described in detail and their functioning has been explained.

I have described the two HL7 standards for the presentation of data in health care. The first standard is a standard demand language designed for the processing of data bases. The second standard is a standardised programming language Arden Syntax, which is used in modules for the decision-making, dissemination of information and the alarming of the medical staff in certain situations (MLM modules).

Key words: project eZdravje, organisation of HL7 standards, language Gello, language Arden Syntax, MLM modules

1. Uvod

Informatika na področju zdravstva je pri nas še zelo slabo izkoriščena in uporabljena. Slovenija, kot članica Evropske unije, je primorana izboljšati stanje na področju komuniciranja, širjenja informacij in povezovanja z evropskimi zdravstvenimi informacijskimi sistemi. Cilj projekta eZdravje je v Sloveniji izboljšati stanje na teh treh področjih ter s tem povečati kakovost življenja njenih državljanov. Zdravstveni kartoni bodo opuščeni in jih bo zamenjal elektronski zdravstveni zapis (EZZ). Tak zapis bo hranil vse ključne informacije o pacientu, njegovem zdravstvenem stanju, prejšnjih diagnozah, stroških zdravljenja, predpisanih zdravilih ter času okrevanja. Hranil bo tudi vse slike pacientovih preiskav. Za hrambo tako velikega števila podatkov bo potrebno vzpostaviti podatkovno bazo, v kateri bodo shranjeni vsi elektronski zdravstveni zapisi, pa tudi podatki o zdravstvenem osebju, stroških poslovanja in upravljanja bolnišnic, itd. Za delo s tako velikimi količinami podatkov, so potrebni standardi, ki jih že uporabljajo v tujini in so se izkazali za zelo uporabne pri delu z zdravstvenimi podatki. Projekt eZdravje je podrobneje predstavljen v drugem poglavju diplomske naloge.

Diplomska naloga se v tretjem poglavju osredotoča na neprofitno standardizacijsko organizacijo HL7, ki postavlja podatkovne standarde v zdravstvu že več kot 22 let. Njihovi standardi poenostavljajo formacijo in izmenjavo zdravstvenih podatkov, omogočajo optimizacijo dela in izboljšano komunikacijo med računalniškimi sistemi. Standarde HL7 uporabljajo po celem svetu in večino je potrdila tudi mednarodna organizacija za standardizacijo (angl. *International Organization for Standardization*). Za pridobitev dostopa do opisov standardov, organizacija HL7 zahteva včlanitev v organizacijo. Slovenija je decembra 2009 postala članica te organizacije in pri projektu eZdravje se bo uporabljal standard HL7 v.3, ki med drugim definira tudi konsistenten podatkovni model, kar posledično omogoča lažjo manipulacijo s podatki, večjo natančnost in hitrejše delo.

V četrtem poglavju sta predstavljena dva standarda za delo z zdravstvenimi podatki, in sicer skriptni jezik Gello in standardni jezik Arden Syntax. Za poizvedbe po podatkovni bazi se uporablja standardni povpraševalni, objektno orientirani skriptni jezik Gello, ki je prilagojen za delo s konsistentnim podatkovnim modelom. Jezik Gello je po delovanju zelo podoben strukturiranemu povpraševalnemu jeziku SQL (angl. *Structure Query Language*), le da uporablja ukaze, ki so prilagojeni izrazom v zdravstvu ter tako omogoča uporabo tudi zdravstvenemu osebju, ki nima veliko znanja o programiranju. Edina razlika med jezikoma Gello in Sql je ta, da Gello podatkov ne more spreminjati, lahko pa izvede kompleksno logiko, ki zdravstvenemu osebju olajša odločanje glede pacientovega nadaljnjega zdravljenja.

Odločitve so v zdravstvu ključnega pomena in vsaka pomoč ali dodatna informacija zdravnikom in ostalemu zdravstvenemu osebju zelo olajša odločanje. Prav s tem namenom je organizacija HL7 ustvarila standardni jezik Arden Syntax, ki se preko modulov uporablja za pomoč pri odločanju, oceni situacije in ob nepravilnostih alarmira pristojno zdravstveno osebje. Prednost modulov imenovanih Medical Logic Modules (MLM) je v tem, da se potem, ko so enkrat spisani in vsebujejo vse potrebne podatke za svojo funkcionalnost, avtomatično aktivirajo ter s tem pomagajo osebju, ki nima nobenega znanja o programiranju. Torej rabijo samo skrbnike, ki jih posodablajo, spreminjajo in izločajo module, ki ne služijo več svojemu namenu ali so zastareli. Avtomatizem je tudi glavna prednost modulov MLM, saj s tem pripomorejo k lažjemu delu s

pacienti, boljšemu razumevanju zdravstvenih težav in nasploh celotnemu vodenju administracije bolnišnic. MLM moduli se uporabljajo v tujini že več kot 30 let in so se v praksi izkazali za zelo uporabne. Jezik Arden Syntax je zelo enostaven za uporabo in zahteva minimalno znanje logike ter programiranja. Ena od prednost MLM modulov je tudi prenosljivost znotraj ene institucije oziroma med institucijami, kar zelo pripomore k njihovi širitvi in uporabi.

2. Projekt eZdravje

Projekt eZdravje [1] predstavlja trenutno enega večjih projektov informatizacije javnih storitev v Sloveniji. Projekt združuje aktivnosti vpeljave rabe komunikacijskih in informacijskih sredstev na področju zdravstva, s katerimi bo mogoče zagotoviti učinkovitejše javno-zdravstvene storitve. Rezultati projekta eZdravje bodo omogočili, da se bo zdravstvena obravnava bolje prilagodila posameznikom, olajšala mobilnost in varnost pacientov, zmanjšalo stroške zdravstvenih storitev, ter izboljšala interoperabilnost v državi in prek meja.

Uvodne aktivnosti projekta eZdravje so se začele v septembru 2008, projekt pa naj bi predvidoma trajal do junija 2015. Za to obdobje je bil pripravljen akcijski načrt in definirani vsi podprojekti. Do tedaj so zagotovljena tudi evropska sredstva, skupaj 27 milijonov evrov od predvidenih 67 milijonov evrov celotnih stroškov. Okvirno je program aktivnosti narejen še do leta 2023, v tem obdobju pa bo treba vse razvite rešitve vzdrževati in nadgrajevati.

Uspeh projekta bo v veliki meri odvisen od zaupanja posameznikov in zdravstvenih delavcev v dvig kakovosti in povečanja varnosti v zdravstvu, ki bo dosežen s projektom eZdravje.

2.1 eZdravje – gonilo informatizacije slovenskega zdravstvenega sistema

Projekt eZdravje pomeni izvajanje konceptov e-Europe in e-Health, ki ju je sprejela EU in predvideva izboljšanje kakovosti življenja prebivalcev EU v zvezi z zdravjem z večjo uporabo komunikacijskih in informacijskih sredstev. Tako EU v svoji strategiji e-zdravja jasno poudarja zahtevo po urejeni zdravstveni informatiki vsake države članice s ciljem povezave evropskih zdravstvenih informacijskih sistemov, ki naj omogočijo uveljavljanje zdravstvenega varstva kjerkoli v EU. To zahteva interoperabilnost zdravstvenih informacijskih sistemov in uporabo enotnih standardov.

Projekt eZdravje je eden prioriteten razvojnih programov v Sloveniji in eden izmed večjih nacionalnih informacijskih projektov, za katerega bo do konca leta 2015 potrebno zagotoviti 67 milijonov EUR. Od tega jih nekaj več kot 27 milijonov zagotavlja Evropski socialni sklad. Za izvedene aktivnosti na projektu je bilo do konca leta 2009 porabljenih 828.964 EUR. Projekt eZdravje tako ni le tehnološka rešitev, temveč gre za nov pristop k vzpostavitvi zdravstva, ki je usmerjen na prebivalce.

Projekt eZdravje bo nudil uporabniku sprva predvsem:

- prilagojene informacije na zdravstvenih portalih s splošnimi in posebnimi informacijami;
- možnost elektronskega naročanja;
- izmenjavo mnenj z zdravstvenimi strokovnjaki preko varne e-pošte;

v nadaljevanju pa tudi:

- uporabo metod telezdravstva oziroma zdravljenja na daljavo;
- oskrbo na domu na daljavo;
- izobraževanje.

Prvi rezultati projekta eZdravje, ki se nanašajo na ureditev nacionalnih čakalnih seznamov, elektronsko naročanje in teleradiologijo, bodo vidni konec leta 2010.

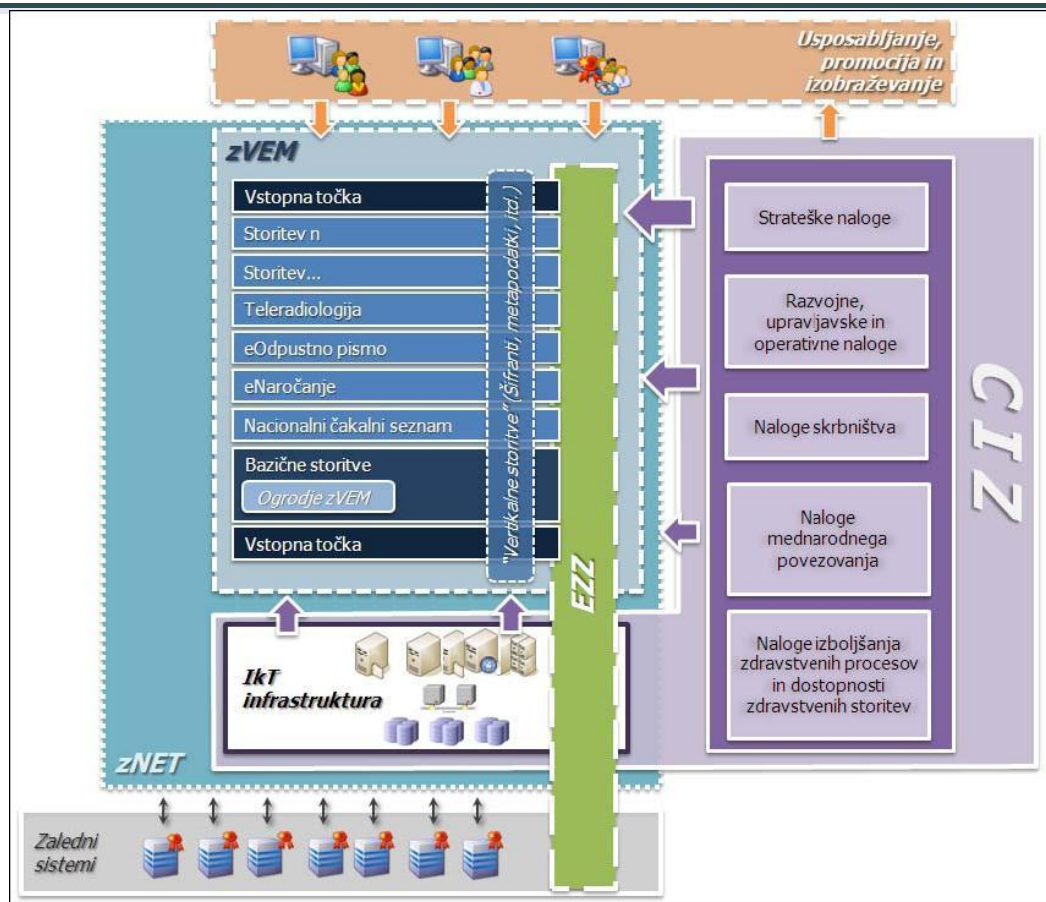
Zaposleni v zdravstvu, ki so vključeni v projekt eZdravje in so preko predstavnikov tudi seznanjeni z njegovo vsebino, že sedaj pričakujejo, da bodo rezultati projekta omogočili celovitejšo in kakovostnejšo obravnavo pacientov ter boljše sodelovanje s kliničnimi specialisti in drugimi zdravstvenimi delavci. Najpomembnejše pa se jim zdi aktivno vključevanje pacientov v procese zdravljenja in njihova večja skrb za lastno zdravje.

2.2 Obseg projekta eZdravje

Kot je bilo že predstavljeno, je projekt eZdravje trenutno eden največjih nacionalnih informacijskih projektov, ki predvideva izvedbo preko 20 podprojektov do leta 2015. Rezultati teh podprojektov bodo omogočali realizacijo ključnih strateških ciljev, ki si jih je zadalo Ministrstvo RS za zdravje, in sicer [2]:

- povečati kakovost in učinkovitost zdravstvenega sistema, kar vključuje lažje načrtovanje in upravljanje zdravstvene organizacije oziroma zdravstvenega sistema kot celote na podlagi kakovostnih in verodostojnih ekonomskih, administrativnih in kliničnih podatkov zdravstvenega sistema;
- mobilizirati ustrezne vire za področje informatike in celovite kakovosti v zdravstvu;
- izboljšati dostopnost zdravstvenih storitev za tiste skupine posameznikov, ki bi bili sicer zaradi svojih zmanjšanih zmožnosti, starosti ali drugih razlogov izključeni;
- uveljaviti e-poslovanje kot običajen način dela v slovenskem zdravstvu.

Celovit pogled na projekt eZdravje prikazuje slika številka 1.



Slika 1 : Celovit pogled na projekt eZdravje [3].

2.3 Pričakovane koristi projekta eZdravje

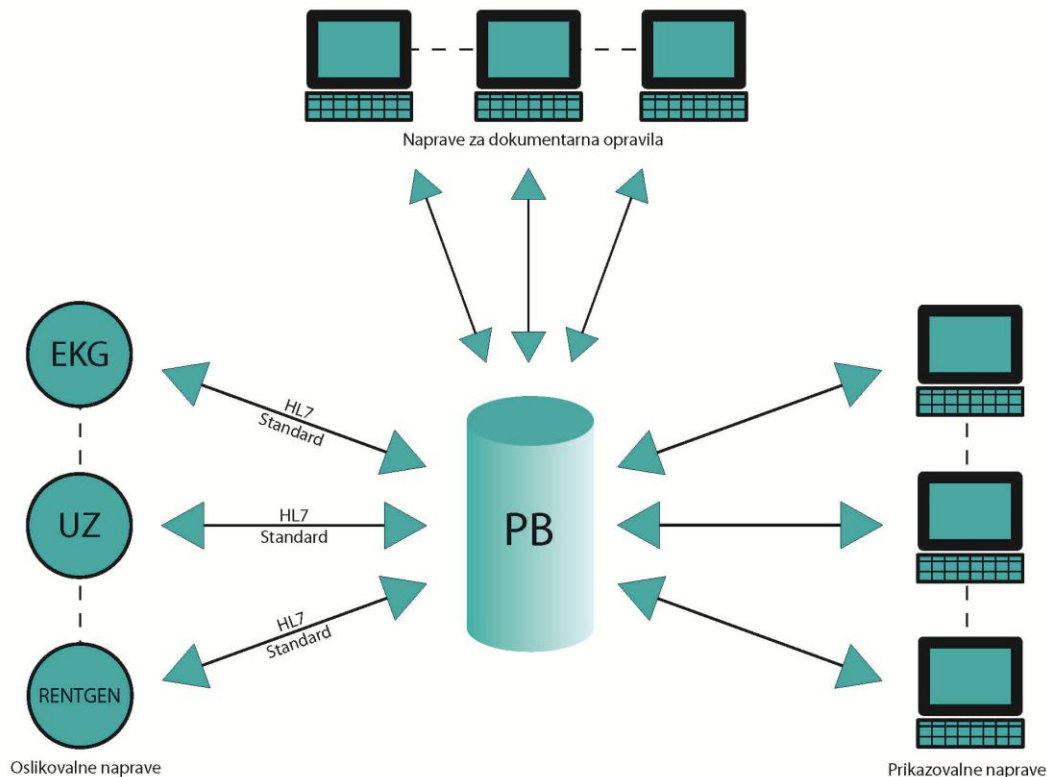
Projekt eZdravje bo s svojimi rezultati zagotovil podlage za sistematično povečevanje učinkovitosti zdravstvenega sistema, izboljšanje pretoka informacij med subjekti znotraj in zunaj zdravstvenega sistema, povečevanje kakovosti dela in življenja vseh ključnih akterjev v zdravstvu, ter učinkovito varstvo pacientovih pravic. Rezultati projekta eZdravje bodo imeli učinke predvsem na:

- zdravstveni sistem kot celoto, saj bodo olajšali izmenjavo informacij med najrazličnejšimi subjekti zdravstvenega sistema;
- posameznike – uporabnike zdravstvenih storitev s pravo, njim prilagojeno informacijo: z zdravstvenimi portali s splošnimi in posebnimi informacijami, z možnostjo elektronskega naročanja, z izmenjavo mnenj z zdravstvenimi strokovnjaki preko varne e-pošte, kasneje pa tudi z uporabo metod telezdravstva (angl. *home tele-health*; telemedicina na domu) in z oskrbo na domu na daljavo (angl. *home telecare*); na ta način bo posameznik prejel storitve za ohranjanje oziroma pridobivanje zdravja v svojem bivanjskem (domačem) okolju ter s tem podpora pri samostojnem življenju;

-
- zdravstvene strokovnjake z dostopom do različnih elementov elektronskega zdravstvenega kartona pacienta, do elektronskih rentgenskih slik in drugih izvidov, z lažjim načrtovanjem postopkov in posegov, s hitrim dostopom do strokovnih virov (medicinske e-knjižnice) in izobraževanja (izobraževanje na daljavo);
 - zdravstvene managerje in upravljavce zdravstvenega sistema, ki so odgovorni za delovanje zdravstvenega sistema, z možnostjo uporabe neosebni strokovnih, organizacijskih in ekonomskih informacij o delovanju zdravstvenega sistema.

2.4 Delovanje elektronskega zdravstvenega informacijskega sistema (eZIS)

Delovanje elektronskega zdravstvenega informacijskega sistema, ki ga bo vzpostavil projekt eZdravje, lahko poenostavljeno opišemo kot veliko bazo podatkov, kamor se bodo shranjevali različni tipi podatkov. Ti podatki se bodo prenašali iz različnih virov (medicinskih naprav, ambulant, laboratorijev, itd.) do centralne baze podatkov preko protokolov, ki ustrezajo HL7 standardom in bodo dostopni vsem pooblaščenim osebam, vključno z osebnimi zdravniki, ki bodo s pomočjo teh informacij hitreje postavili pravo diagnozo. Vsak pacient bo imel svoj elektronski zdravstveni zapis (EZZ), ki bo predstavljal le enega izmed mnogih zapisov v centralni bazi podatkov. V tej »osebni bazi« bodo shranjeni vsi osebni podatki uporabnikov zdravstvenih storitev, njihove prejšnje diagnoze, EKG slike, rentgenske slike, načini zdravljenja, itd. Na ta način bo lahko pacient prišel do zdravnika, ki ni njegov osebni zdravnik in vseeno prejel ustrezno zdravstveno oskrbo in to brez prenašanja njegovega zdravstvenega kartona.



Slika 2 : Prikaz komunikacije med vozlišči v zdravstvenem informacijskem sistemu.

2.5 Podporni standardi v projektu eZdravju

V elektronskem zdravstvenem informacijskem sistemu Slovenije, ki bo rezultat projekta eZdravje, se bodo uporabljali naslednji standardi [4]:

- ISO/CEN 13606 in ISO/CEN 21549 (standarda, ki določata sestavo elektronskega zdravstvenega zapisa (EZZ) pacienta);
- OpenEhr (odprti standard, ki opisuje, kako se urejajo, shranjujejo, prejemajo in izmenjujejo podatki, shranjeni v EZZ [5]);
- HL7 v.3 (standard za izmenjavo podatkov, ki bo natančneje opisan v nadaljevanju).

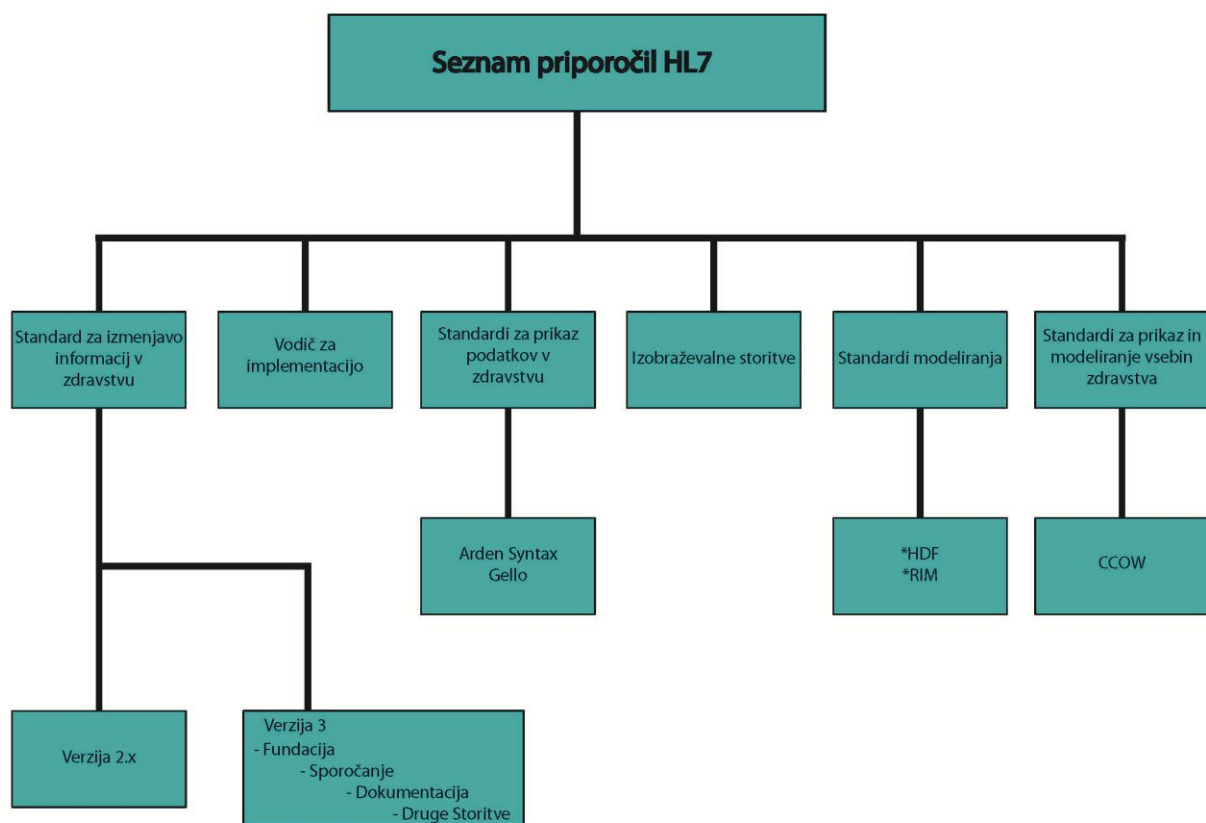
3. Organizacija Health Level Seven

V nadaljevanju bo predstavljena organizacija Health Level Seven in pregled njenih standardov.

3.1 Predstavitev organizacije Health Level Seven

Neprofitna organizacija Health Level Seven International (HL7) [6] je bila ustanovljena leta 1987 in se ukvarja z razvojem ANSI-akreditiranih standardov, ki so namenjeni izmenjavi, povezovanju in iskanju elektronskih zdravstvenih podatkov, ki služijo kot podpora upravljanju in vrednotenju zdravstvenih storitev. HL7 torej ne razvija programske opreme, temveč postavlja standarde, ki omogočajo izmenjavo zdravstvenih informacij med različimi sistemi in zagotavlja celovito ogrodje za izmenjavo in obdelavo informacij.

Organizacija ima preko 2.300 aktivnih članov, med katerimi je približno 500 predstavnikov podjetij, ki skupaj pokrivajo več kot 90% prodajalcev informacijskih sistemov v zdravstvenem sektorju.



* Predhodnika standarda verzije 3

Slika 3 : Prikaz priporočil HL7.

"Level Seven" (sedmi nivo) v imenu organizacije se nanaša na najvišji nivo sedemstopenjskega komunikacijskega modela, imenovanega Open Systems Interconnection (OSI) [7]. Sedmi nivo se torej nanaša na aplikacijski nivo tega modela (glej sliko 4) in je namenjen neposrednemu prenosu spletnih storitev in procesov ter vsebuje podporne in aplikacijske protokole. Organizacija HL7 si prizadeva za razvoj stroškovno učinkovitega pristopa k sistemski povezanosti.

Ministrstvo za zdravje Slovenije je z decembrom 2009 postalo član organizacije Health Level Seven. Vključitev in podpora vladnih institucij in podjetij, ki razvijajo informacijske sisteme na področju zdravstva, sta ključnega pomena pri nadaljnji širitvi in večanju pomena standarda HL7 ter pri splošnem uspehu organizacije.



Slika 4 : Prikaz nivojev v OSI modelu [6].

3.2 Dosežki organizacije HL7 in njeno delovanje

Najpogosteje uporabljene HL7 specifikacije so standardi sporočanja, ki omogočajo različnim aplikacijam v zdravstvu, izmenjavo ključnih zdravstvenih in administrativnih podatkov. HL7 verzija 2.x standard sporočanja je, po navedbi vira, najbolj razširjen standard v zdravstvu na svetu. Leta 2009 je bil ta standard objavljen kot ISO [8] standard. V ZDA je standard verzije 2 uporabljen v večini zdravstvenih ustanov. Standard HL7 verzije 3 uporabljajo vladne agencije v ZDA, kot sta na primer Uprava za prehrano in zdravila (Food and Drug Administration) in Oddelek za veterane (Department of Veterans Affairs). Različica 3 se veliko uporablja tudi v Kanadi, Veliki Britaniji, Nizozemski, Nemčiji in Mehiki.

Podružnice organizacije HL7 so razširjene v več kot 30-ih državah po vsem svetu, med katerimi so Argentina, Avstralija, Avstrija, Brazilija, Kanada, Čile, Kitajska, Kolumbija, Hrvaška, Češka, Danska, Finska, Francija, Nemčija, Grčija, Hong Kong, Indija, Irska, Italija, Japonska, Koreja, Nizozemska, Nova Zelandija, Romunija, Rusija, Singapur, Španija, Švedska, Švica, Tajska, Turčija in Urugvaj. Na tem spisku je tudi naša država [9], čeprav na spletni strani HL7 [6] ni navedena.

3.3 Pregled standardov HL7

HL7 določa standarde interoperabilnosti, ki izboljšujejo izmenjavo podatkov, optimizacijo dela, zmanjšujejo dvomnosti in krepijo prenos znanja med vsemi interesnimi skupinami, vključno z izvajalci zdravstvenih storitev, državnimi agencijami, proizvajalci zdravstvenih pripomočkov in aparaturo ter bolniki.

3.3.1 Standard sporočanja HL7 verzija 2.x

Standard sporočanja HL7 verzija 2.x [11, 12] so razvili za elektronsko izmenjavo podatkov na področju zdravstva in je, po navedbi vira, najbolj razširjen standard v zdravstvu na svetu. Od leta 1987, ko so razvili prvo različico standarda, pa do danes je bilo izdanih sedem različic tega standarda. Vse novejšje različice 2.x so kompatibilne s prejšnjimi različicami, saj omogočajo raven uporabe, ki preze vse neprepoznane dele sporočila. To pomeni, da lahko tudi starejše aplikacije sprejemajo sporočila, poslana iz novejših aplikacij, ki imajo več podatkov, in to brez težav, ki bi nastale zaradi neprepoznanih podatkov in posledično javljanja napak.

Standard zajema sporočila za izmenjavo informacij na naslednjih področjih:

- demografski podatki pacienta;
- stroški obravnave pacienta;
- zavarovanje pacienta;
- diagnoze pacienta;
- sprejem pacienta v bolnišnico, odpust iz bolnišnice ter premeščanja med bolnišnicami;
- naročanje pacientov na preiskave, storitve in postopke;
- poročanja o rezultatih preiskav;
- sinhronizacija datotek med sistemi;
- upravljanje z zdravstveno dokumentacijo;
- razporejanje pacientov na preiskave in po diagnostično-terapevtskih napravah.

3.3.2 Standard sporočanja HL7 verzija 3.x

Standard HL7 V3 [11, 12] je bil izdan konec leta 2005. Pri njegovem razvoju so imeli veliko več vpliva ameriška vlada in uporabniki zdravstvenih informacij kot pa strokovnjaki za izdelavo zdravstvenih vmesnikov. Standardi V3 namreč niso kompatibilni s standardi V2, tako da obstoječi V2 vmesniki ne bodo sposobni komunicirati (brez večjih modifikacij) z vmesniki, ki uporabljajo standard V3. Standard definira serija sporočil (imenovanih interakcije), ki podpirajo vse postopke v zdravstvu. Sporočila HL7 V3 so zasnovana na XML sintaksi kodiranja.

Standard V3 je bil ustvarjen za odpravo nekaterih pomanjkljivosti, ki so se pojavljale pri standardu V2. Te pomanjkljivosti so:

- nekonsistenten in nepredvidljiv podatkovni model;
- odsotnost formalne metodologije s podatki o modeliranju, ustvarjanje nedoslednosti in težav pri razumevanju;
- pomanjkanje jasno določenih vlog za zahteve in sporočila, ki so uporabljena pri različnih zdravstvenih funkcijah;
- prevelika prožnost in nepopolne rešitve.

Cilji standarda V3 so bili:

- sprejetje standarda po vsem svetu,
- definiranje novega konsistentnega podatkovnega modela;
- izdelava standarda z večjo natančnostjo in manj nejasnostmi.

Odločitev, da bo V3 nov standard, ki ne bo kompatibilen z V2, je kriva, da standard danes še ni tako široko uporabljen kot njegov predhodnik.

Za prehod na V3 mora uporabnik ustvariti in vzdrževati V2-zasnovan vmesnik, ki poganja V2 aplikacije, hkrati pa implementirati vmesnik za poganjanje V3 aplikacij. Zato se V3 uporablja samo v aplikacijah, ki niso odvisne od V2 komunikacije, v aplikacijah, ki niso pogojene z zgodovino komuniciranja, in v ustanovah, v katerih ameriška vlada vztraja pri njihovi uporabi.

3.3.3 Standard CCOW

CCOW (Clinical Context Object Workgroup) [11, 12] je standard, namenjen upravljanju z zdravstvenimi podatki in posredno izboljšanju delavnih procesov. Omogoča enkratni vpis, kar pomeni, da se lahko zdravstveni delavec prijavi s svojim uporabniškim imenom in geslom v več aplikacij in tako dobi dostop do različnih pacientovih zdravstvenih podatkov hkrati. Standard je sprejela organizacija ANSI septembra leta 2000 (različica 1.2).

3.3.4 Standard CDA

CDA (Clinical Document Architecture) [11, 12] je standard, ki je zasnovan na XML jeziku in je namenjen predstavitvi in prenosu zdravstvenih podatkov. CDA definira strukturo in semantiko zdravstvenih dokumentov (na primer odpust pacienta iz bolnišnice, napredek zdravljenja, itd.) ter jih pripravi na izmenjavo preko mreže.

CDA dokument je strogo definiran objekt, ki lahko vsebuje tekst, slike, zvok in druge multimedijske vsebine. Dokumenti se lahko prenašajo s poslanimi sporočili ali pa obstajajo samostojno zunaj njega. Standard CDA verzije 2.0 je sprejela organizacija ANSI maja leta 2005.

4. Standardi HL7 za izmenjavo podatkov v zdravstvu

V nadaljevanju bosta predstavljena standardni povpraševalni jezik Gello in standardni jezik Arden Syntax.

4.1 Standardni povpraševalni jezik GELLO

4.1.1 Uvod

GELLO [10, 11, 12, 13] je standardni, od platforme odvisni skriptni jezik, ki ga lahko uporabljajo vse aplikacije v zdravstvu, ki zahtevajo poizvedovanje in sklepanje o podatkih ter oceno logičnih izrazov. GELLO je povpraševalni jezik za delo s podatkovnimi bazami, ki je zelo podoben jeziku SQL.

Izdelava skriptnega jezika GELLO se je začela leta 2001, javnosti pa je bil predstavljen v letu 2005. Jezik sta sprejeli kot mednarodni standard organizaciji ANSI in HL7 in se uporablja za jezikovno podporo pri odločanju.

GELLO je namenjen uporabi:

- zdravnikom;
- raziskovalcem;
- epidemiologom;
- sistemskim inženirjem;
- programerjem;
- ekonomistom v zdravstvu.

GELLO omogoča pomoč pri izbiri med množico podatkov (kot je na primer celoten nabor podatkov o pacientu) in tako lajša odločitve zdravstvenim delavcem. GELLO uporablja abstrakten "navidezni zdravstveni zapis", tako da lahko ista GELLO koda teče na več različnih sistemih in uporablja različne tipe shranjenih podatkov. Sam jezik ne more spreminjati zdravstvenega zapisa, lahko pa izvede kompleksno logiko odločanja o pacientovem zdravju. Uporablja se lahko tudi za pridobivanje podatkov, ki so namenjeni odločitvam v drugih naprednejših aplikacijah in za ustvarjanje zapletenih nizov podatkov za grafikone ali statistične analize.

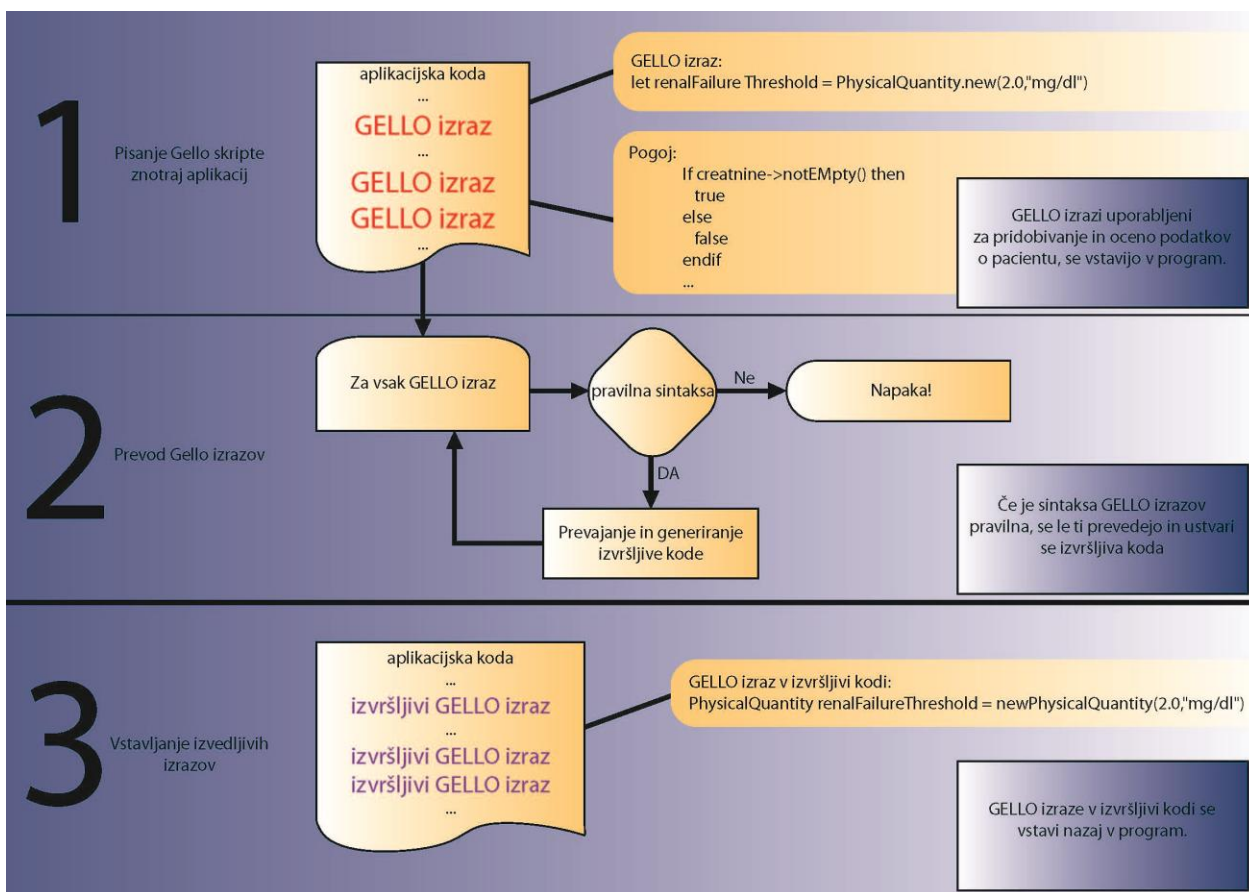
Da bi Gello jezik služil svojemu namenu, mora zadostiti naslednjim pogojem:

- sintaksa jezika mora biti objektivno orientirana (tak pristop dopušča prilagajanja, inkapsulacijo ter razteznost; zaradi tega je poenostavljeno vrednotenje zdravstvenih podatkov);

- sposobnost dostopanja in manipuliranja s podatki danega podatkovnega modela;
- izrazi znotraj jezika morajo znati pridobivati informacije iz različnih podatkovnih virov, graditi kriterije odločanja ter biti sposobni logičnega odločanja glede določene vrednosti ali podatka;
- jezik mora omogočati manipuliranje s časovnimi podatki (ura, dan, mesec, leto).

4.1.2 Predstavitev jezika Gello

Gello program je pravzaprav skripta, ki jo prevajalnik spremeni v paket z izvršljivo kodo. Programerji se morajo držati določenih pravil pisanja, ki jim pravimo sintaksa jezika, ter upoštevati pravila semantike.



Slika 5 : Prikaz delovanja jezika Gello

Primer: Za pacienta iščemo zdravilo, z nivojem kalcija in fosforja pod določeno vrednostjo. Visoki nivo Ca-PO₄ (kalcijevega fosfat) v krvi lahko povzroči odpoved ledvic ter tendenco h kalcifikaciji mehkih tkiv. V jeziku GELLO bi povpraševanje izvedli kot sledi.

V prvem delu našega programa razberemo iz pacientovih podatkov najvišje izmerjene nivoje kalcija, fosforja in kreatina (ta nas zanima zaradi nevarnosti odpovedi ledvic), ter te vrednosti dodelimo spremenljivkam, ki jih kasneje uporabimo.

```
let lastCreatinine : Observation = Observation→ select (code=
Factory.CodedValue("SNOMED-CT", "xxxxxx")).sortedBy(effectiveTime.high).last()

let lastCalcium : Observation = Observation→ select (code =
Factory.CodedValue("SNOMED-CT", "yyyyy")).sortedBy(effectiveTime.high).last()

let lastPhosphate : Observation = Observation→ select (code=
Factory.CodedValue("SNOMEDCT", "zzzzz")).sortedBy(effectiveTime.high).last()
```

V nadaljevanju postavimo prag odpovedovanja ledvic.

```
let renal_failure_threshold : PhysicalQuantity = Factory.PhysicalQuantity(
"2.0, mg/dl")
```

Postavimo prag vsebnosti Ca-PO₄ (kalcijevega fosfata).

```
let threshold_for_osteodystrophy : int = 70
```

Pogoj nam vrne odgovor, ali obstaja nevarnost odpovedi ledvic. Znaka <>, napisana skupaj, v jeziku GELLO pomenita neenačaj.

```
let renal_failure : Boolean = if lastCreatinine <> null and
lastCreatine.value.greaterThan(renal_failure_threshold)
then
true
else
false
endif
```

Sedaj preverimo nivoja kalcija in fosforja ter ju pomnožimo. Če ima kateri od njiju »null« vrednost, program vrne -1.

```
let calcium_phosphate_product : real = if lastCalcium
<> null and lastPhosphate <> null
then
lastCalcium.value * lastPhosphate.value
else
-1
endif
```

Za konec samo še prej pomnoženo vrednost kalcijevega fosfata primerjamo s prej navedenim pragom.

```
if renal_failure and calcium_phosphate_product >
threshold_for_osteodystrophy then
"whatever action or message"
else
"whatever action or message"
endif
```

4.1.2.1 Komentiranje kode

Jezik Gello ima dva načina komentiranja kode, in sicer:

- komentar ene vrstice (`//koda vrstice`);
- komentiranje več vrstic kode (`/* koda vrstice
koda vrstice*/`).

4.1.3 Podatkovni tipi jezika Gello

Vsak Gello podatek mora imeti vrednost in tip. Gello pozna enostavne podatkovne tipe, tipe tuple (posebnost Gella) ter zbirke.

4.1.3.1 Enostavni podatkovni tipi

Enostavni podatkovni tipi predstavljajo osnovni del vsakega podatka, s katerim lahko program operira. Gello uporablja naslednje tipe:

- `integer` (cela števila, ki so lahko pozitivna in negativna; zavzemajo lahko vrednosti od $-(2^{53}-1)$ pa do $2^{53}-1$);
- `real` (realna števila; zavzemajo lahko realne številske vrednosti od $-(2^{63}-1)$ pa do $2^{63}-1$);
- `string` (niz znakov; vsebujejo znake razširjene ASCII tabele);
- `boolean` (pogoj; zavzema lahko vrednosti `true`, `false` in `unknown`).

4.1.3.2 Zbirke

Zbirke so sezname vrednosti, ki so enakega podatkovnega tipa. Elementi v zbirki so lahko preprostega tipa (`integer`, `string`, itd.) ali bolj kompleksni podatki kot so zbirke in celo razredi. V Gello jeziku obstajajo tri vrste zbirk:

- zbirka `Set`, v kateri ne sme priti do podvajanj elementov;
- zbirka `Bag`, v kateri so elementi lahko iste vrednosti;
- zbirka `Sequences`, ki je podobna zbirki `Bag`, le da je pomembno tudi zaporedje elementov.

Primeri zbirk v GELLO jeziku:

Preprosta zbirka `set` s tremi nizi:

```
Set{ "apple", "orange", "pear" }
```

Preprosta zbirka `Bag` s celimi števili, kjer so nekatere vrednosti podvojene:

```
Bag{ 1, 2, 2, 3, 4, 4, 4, 5 }
```

Preprosta zbirka `Sequences`, kjer je pomembno zaporedje elementov:

```
Sequence{ 1, 2, 3, 2, 4, 5, 4, 4 }
```

Gello jezik je zelo močno orodje ravno zaradi uporabe zbirk. Običajno ima podatkovni model številne zbirke, ki se razvrščajo po svojih vrednostih. Za lažje delo z zbirkami in ustvarjanje novih zbirk, imamo tako imenovane operatorje zbirk. Med najbolj uporabnimi sta operatorja `select` in `collect`.

`Select` nam ustvari novo podmnožico zbirke, ki ustreza pogojnemu stavku.

Primer uporabe:

```
context observations: Observations from Model.observations  
  
let sodiums = observations->select(code.name = "Sodium")  
  
sodiums.value
```

V našem primeru imamo zbirko `observations`, ki ima v danem modelu kot element vrednost razreda `Observation`. Eden od elementov zbirke se imenuje `code` in pripada razredu `CodedValue`. S `select` operatorjem preverjamo vrednost elementov in če ta ustreza nizu "Sodium", se ta element izbere in doda v novo zbirko. Na podoben način delujejo tudi ostali operatorji zbirk.

4.1.3.3 Tip Tuple

Tip `Tuple` je podoben zbirkam, le da so elementi znotraj njega povezani in ni nujno, da so podatki enake vrste. Vsak element v tipu `tuple` se kliče po imenu in ima svoj edinstven tip elementa.

V našem primeru bomo z navedbo `let` deklarirali tip `tuple`, ki bo v našem primeru vseboval podatke o pacientu. `Tuple` tipi so lahko vgnezdjeni v drugih kompleksnih podatkovnih tipih in obratno. Deklaracija tipa `tuple` z imenom `patient_contact`, ki ji sledi inicializacija je videti takole:

```
let patient_contact:  
  
tuple( surname: string,  
  
       givenname: string,  
  
       streetnumber: integer,
```

```
streetname
city: string,
zipcode: integer
country: string)
= tuple{
    surname = "Smith",
    givenname = "Fred",
    streetnumber = 123,
    city = "MoTown",
    zipcode = 998877,
    country = "Republic of MoTownomia"}
```

4.1.4 Operatorji

Operatorji v jeziku Gello se delijo v dve skupini, in sicer v enostavne (operacije se izvršijo samo na enem operandu) in v binarne (operacije potekajo med dvema operandoma). Nadalje se operatorji v jeziku Gello delijo na preproste operatorje, operatorje, ki se izvajajo nad razredi, in operatorje, ki se izvajajo nad zbirkami.

4.1.4.1 Preprosti operatorji

Preproste operatorje delimo na dve skupini, in sicer:

- aritmetične operatorje (ti operatorji delujejo na realnih in celih številih):
 - (+) seštevanje dveh operandov;
 - (-) odštevanje dveh operandov;
 - (*) množenje dveh operandov;
 - (/) deljenje dveh operandov;
 - (-) negacija operanda;

 - (div) celi del števila pri deljenju;
 - (mod) ostanek pri deljenju;

- operatorji `boolean` oziroma Boolovi operatorji (ti operatorji samo na tipih `Boolean` v jeziku Gello pa se lahko uporabljajo tudi za neznane vrednosti):
 - (and) logični `in` med dvema operandoma;
 - (or) logični `ali` med dvema operandoma;
 - (xor) logični `ekskluzivni ali` med dvema operandoma;
 - (not) negacija operanda.

Tabela 1: Pravilnostna tabela za Boolove operatorje.

A	B	A and B	A or B	A xor B	not A
false	false	false	false	false	true
false	true	false	true	true	true
true	false	false	true	true	false
true	true	true	true	false	false
false	unknown	false	unknown	unknown	true
true	unknown	unknown	true	unknown	false
unknown	false	false	unknown	unknown	unknown
unknown	true	unknown	true	unknown	unknown
unknown	unknown	unknown	unknown	unknown	unknown

4.1.4.2 Operatorji razredov

Operatorji, ki se izvajajo nad razredi, so trije, in sicer:

- `value.isUndefined()` vrne `true`, če je vrednost `null` ali nedefinirana;
- `value.isDefined()` vrne `true`, če vrednost ni `null` ali nedefinirana;
- `value.isTypeof(name)` vrne `true`, če ima razred vrednost danega imena.

4.1.4.3 Operatorji zbirke

Vse operacije, ki se vršijo nad zbirkami v Gello, se prožijo z znakom `→`. Torej puščica loči operacije, ki se vršijo samo nad zbirkami, od ostalih operacij. Klicanje operacije z želenim operandom napišemo takole:

```
zbirka → operatorZbirke(parametri)
```

Primer:

```
let sodiums = observations -> select(code.name = "Sodium")
```

V danem izrazu deklariramo novo zbirko `sodiums`, ki vsebuje elemente tipa `observations`, vendar samo tiste, ki imajo atribut `code` z imenom `»Sodium«`.

Operatorji zbirke lahko uporabljajo po en pogoj ali več za parameter, medtem ko drugi uporabljajo števila ali pa so kar brez njega. Gello ima veliko število vnaprej definiranih operatorjev zbirke, tako da bodo v nadaljevanju opisane samo najbolj uporabne. Operator `select` (`boolean izraz`) smo spoznali v prejšnjem primeru. Operator `reject` (`boolean izraz`) deluje ravno nasprotno in izbere vse tiste elemente, ki ne ustrezajo parametru. Med pogosto uporabljene operatorje zbirke spadajo še:

- `collect` (`izraz`) operator `collect` se prehodi čez zbirko in za vsak element izračuna vrednost izraza in dobljene vrednosti zapiše v novo zbirko;
- `forall` (`boolean izraz`) operator vrne `true`, če vsi elementi v zbirki ustrezajo danemu pogoj;
- `exists` (`boolean izraz`) vrne `true`, če vsaj en element ustreza pogoju;
- `count` (`objekt`) operator vrne celo število, ki pove, kolikokrat se je objekt pojavil v zbirki;
- `includesAll` (`parametri zbirke`) vrne `true`, če so vsi podani parametri prisotni v zbirki.

Za primer uporabe operatorja zbirke `includesAll` navedimo stavek v jeziku Gello, ki bo omogočil ugotoviti, ali sta med pacientovimi simptomi, shranjenimi v zbirki `problemList`, navedena vročina in vnetje. To napišemo takole:

```
patient.problemList → collect(code) → includesAll(Set{"vročina", "vnetje"}).
```

4.1.5 Razredi

Razredi so podobni tipom tuple in imajo imenovane elemente, ki se jim reče atributi. Razredi pa imajo definirane tudi operacije, ki se nad njimi izvajajo. V splošnem imajo podatkovni modeli, ki jih uporabljajo programi Gello, številne razrede, ki predstavljajo komponente tega razreda.

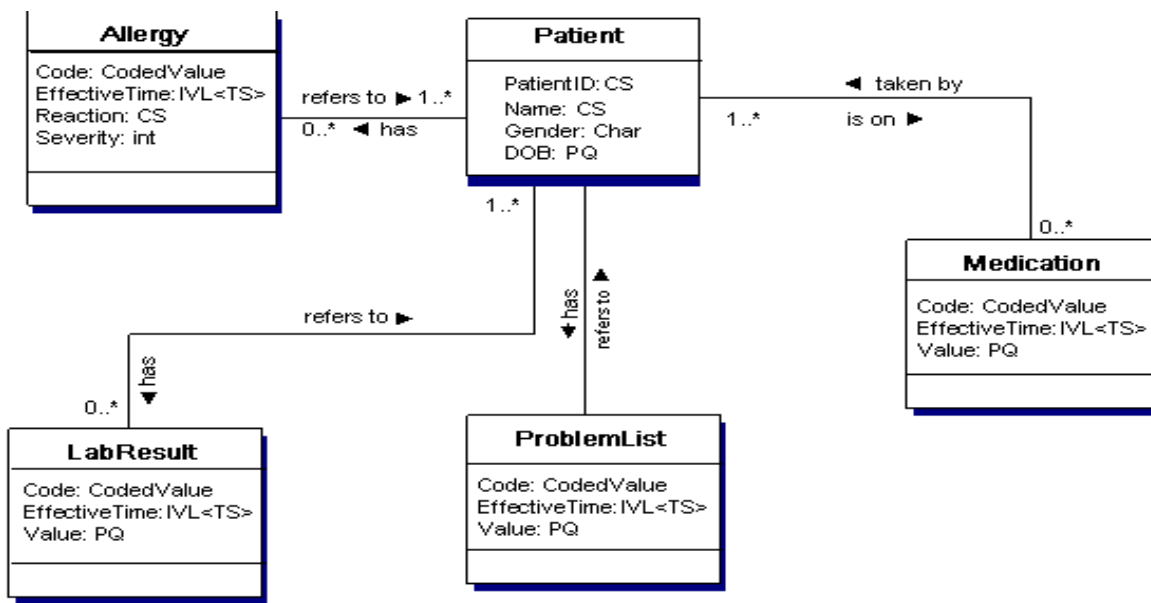
Vsak razred ima attribute, na katere se sklicujemo preko operanda “.” in z imenom atributa. Torej, če imamo spremenljivko z imenom `obs` razreda `Observation`, ki ima atributa `ime` in `starost`, se lahko na njiju sklicujemo z ukazom `obs.ime` in `obs.starost`. Lahko pa se sklicujemo tudi na atribut atributa. Na primer ukaz `obs.ime.priimek` nam vrne priimek pacienta.

Operacijam, ki se izvajajo nad razredi, lahko dodeljemo vrednosti, ki so podobne atributom, le da vračajo rezultat operacije z uporabo naše vrednosti. Operacije so enake kot metode v drugih objektno usmerjenih jezikih.

Primer uporabe operacije nad razredom: imamo instanco razreda `Patient`, ki jo poimenujemo `patient` in vsebuje operacijo imenovano `prescription_count_for_recent_years(n)`, ki nam poda število zdravil, ki so bili predpisani pacientu v `n` letih. Torej s klicem operacije

`patient.prescription_count_for_recent_years(1)` dobimo število predpisanih zdravil v zadnjem letu.

Podatkovni model v Gellu je sestavljen iz petih razredov, ki imajo svoje attribute in lastne operacije, ki se nad njimi izvajajo. Glavni razred `Patient` je z ostalimi štirimi povezan preko vnaprej določenih operacij, ki kličejo podatke iz drugega razreda.



Slika 6 : Podatkovni model v Gellu [13].

4.2 Standardni jezik Arden Syntax

V nadaljevanju bo predstavljen standardni jezik Arden Syntax.

4.2.1 Uvod

Arden Syntax [14, 15, 16] je standardiziran programski jezik, ki se uporablja za predstavitev in izmenjavo zdravstvenih podatkov preko modulov, imenovanih Medical Logic Modules (MLM). Vsak MLM vsebuje vzdrževalne informacije, povezave do drugih virov informacij, ter dovolj logike, da naredi logično odločitev. Ker ti moduli niso enostavni za uporabo in so si nekateri tudi konfliktni, si uporabniki jezika prilagodijo module svojemu delovnemu okolju. Arden Syntax je bil predstavljen leta 1989, vendar sta ANSI in HL7 sprejela šele verzijo 2.0 leta 1999.

Programski jezik ni polno funkcionalen, saj ne vsebuje kompleksnih struktur, module zanj pa naj bi vzdrževalo zdravstveno osebje z majhnim znanjem programiranja ali celo brez njega. Prednosti programskega jezika so:

- standardiziran način integracije znanja v bolnišnični informacijski sistem;
- preprostost, saj je namenjen uporabi osebjem brez znanja programiranja;
- omogoča dostop do podatkov, sproža dogodke in pošilja sporočila želenemu uporabniku; jasno definira povezave med zdravstvenimi podatki ter načine klicanja modulov preko dogodkov;
- nudi zajetno podporo časovnim funkcijam, saj vsi zdravstveni podatki vključujejo čas, ko se neki dogodek zgodi; Arden Syntax poskrbi, da se vsak dogodek zabeleži in da dobi časovni žig, saj je čas pomemben zdravstveni podatek.

4.2.2 MLM moduli

Moduli MLM predstavljajo način odločanja zdravnikov v logični obliki. Modul vsebuje tekst, informacije ter kodo, napisano v vnaprej določeni sintaksi in skupaj shranjeno v ASCII tekstovni datoteki (ANSI X3.4 - 1986). Taka datoteka lahko vsebuje več modulov MLM, ki so ločeni s presledkom ali komentarjem. Prednosti modulov so:

- uspešno opravljajo svojo nalogo že več kot trideset let in se še vedno razvijajo ter nadgrajujejo;
- vse informacije, ki se nahajajo v moduli MLM in so napisane v Arden Syntax, so prenosljive preko modulov samih in prilagojene okolju, v katerem se uporabljajo (bolnišnični informacijski sistem).

Slabosti pa so:

- moduli MLM imajo omejitve pri prikazovanju vgnezenih ukazov; v praksi to pomeni, da če en modul izračuna neko vrednost, drugi modul do nje ne more dostopati in jo mora ponovno izračunati;
- ker ni standardnega podatkovnega modela, ki bi bil prilagojen pacientovim in bolnišničnim podatkom, je delitev informacij preko modulov omejena; torej si lahko vsaka bolnišnica prilagodi module, kot njej ustreza, vendar se s tem izgubi prenosljivost.

Sledi primer modula MLM, na katerem so preko komentarjev opisani glavni deli (ti pa podrobneje v naslednjem poglavju):

```
maintenance:      //vzdrževalni del modula  
  
title: Agranulocytosis and  
       Trimethoprim/Sulfamethoxazole;
```

```

filename: anctms;
version: 2.00;
      institution: Zoo University;
author: Dr. Bonzo;
specialist: ;
date: 7/20/1989, 7/23/1990;
validation: testing;

library: //knjižni del modula

      purpose: Display the Arden Syntax;
keywords:
      granulocytopenia; agranulocytosis;
      trimethoprim; sulfamethoxazole;
citations:
      1. Anti-infective drug use in
      relation to the risk of
      agranulocytosis and aplastic
      anemia.... Arch Int Med
      1989;149 (5) :1036-40.
links:
      MeSH agranulocytosis/ci and
      sulfamethoxazole/ae;

knowledge: //logični del modula

type: data-driven;

data:
      /* neutrophile count in #/mm3 */

      anc := read last 2 from ({query //poenostavljeno klicanje EZZ
      for ANC) where it occurred //dobimo pa zadnji dve
      within the past 1 week); //vrednosti belih krvničk v krvi

      pt_taking_tms := read exist //EZZ nam pove ali pacient jemlje
      (query for TMS order); // določena zdravila

evoke: on storage of (ANC); //na podlagi nivoja belih krvničk se modul
      //poda v logično odločanje
logic: //logični del modula -vsebuje vso potrebno logiko

      if pt_taking_ttms /*1*/ //v pogojnem stavku primerja vrednosti
      and last anc < 1000
      and decrease of anc > 0 then
      conclude true //ter poda zaključek
else
      conclude false;

action: store "Caution: The patient's //Če modul zaključi z
relative granulocytopenia may be //vrednostjo true, vrne
exacerbated by //opozorilo
trimethoprim/sulfamethoxazole.";

```

Opozorilo nam pove, da je možni krivec nizke ravni belih krvničk pri pacientu jemanje zdravila trimethoprim ali sulfamethoxazole.

4.2.2.1 Struktura modulov MLM

Vsak modul je sestavljen iz treh delov, ki si morajo slediti v naslednjem vrstnem redu:

- vzdrževanje (*maintenance*);
- knjižnica (*library*);
- znanje (*knowledge*).

Moduli se zaključijo z ukazom `end` in podpičjem.

4.2.2.2 Vzdrževalni del modula

Vzdrževalni del modula vsebuje podatke, ki niso povezani z zdravstvenimi podatki, temveč se uporabljajo za vzdrževanje in spreminjanje informacij o samem modulu. V tem delu se tudi zapiše, katera verzija Arden Syntax se uporablja. Vzdrževalni del je zelo obsežen, zato navajam samo obvezne dele, ki jih mora imeti vsak modul, ki ga Arden Syntax uporablja. Ti so:

- naslov – ta nam na kratko pove, čemu služi ta modul; na primer, imamo modul, ki preverja količino antigenov za hepatitis B pri nosečnicah; koda bi bila videti takole:

```
title: Hepatitis B Surface Antigen in Pregnant Women;;
```

- ime modula – predstavlja niz dolg od 1 do 80 črk in omogoča enolično identifikacijo modula; rezervirana beseda za ime je `mlmname` in mora biti drugačna, kot je ime datoteke, v kateri je modul:

```
mlmname: hepatitis_B_in_pregnancy;;
```

- verzija – tukaj ne gre za verzijo Arden Syntax (ta podatek ni obvezen), temveč za verzijo modula; priporočljivo je, da se začne z verzijo 1.0 in se ob majhnih spremembah povečuje za 0.1, ob večjih spremembah pa za 1:

```
version: 1.00;;
```

- institucija – tukaj navedemo podatek, v kateri instituciji je modul nastal:

```
institution: Columbia University;;
```

- avtorji – spisek avtorjev, ki so sodelovali pri izdelavi modula; vsakega posebej se napiše v svojo vrstico po vrstnem redu z imenom, priimkom, stopnjo izobrazbe ter opcijsko še elektronskim naslovom:

```
author: John M. Smith, Jr., M.D. (jms@camis.columbia.edu);;
```

- skrbnik – oseba v instituciji, ki je odgovorna za validacijo in inštalacijo modula v sistem; ob prenašanju modulov iz ene institucije na drugo pustimo ta del prazen, saj mora institucija, ki je sprejela modul od druge, sama navesti to osebo in s tem prevzeti odgovornost nase:

```
specialist: Jane Doe, Ph.D.;;
```

ali

```
specialist: ;;
```

- datum – tukaj navedemo datum zadnje verzije modula v obliki leto–mesec–dan:

```
date: 2010-01-02;;
```

- validacija – ta nam pove stanje modula; tukaj imamo štiri opcije, in sicer:

1. produkcija (odobreno za uporabo v zdravstvu);
2. razvoj (faza izdelave modula);
3. testiranje (faza testiranja modula);
4. zastaran (v zdravstvu se ne uporablja več).

Če bi želeli označiti naš modul za testiranje, bi to napisali takole:

```
validation: testing;;
```

Moduli se v stanju produkcije nikoli ne delijo z drugimi institucijami, saj mora to storiti dodeljeni skrbnik.

4.2.2.3 Knjižnični del modula

Knjižnični del vsebuje podatke, ki se nanašajo na vsebino modula MLM, to je na razlago in do internetnih povezav, ki se nanašajo na tematiko modula. Ti podatki zelo olajšajo iskanje informacij o določeni bolezni, zdravljenju ter poenostavijo povezovanje med moduli. Ker je knjižnični del sestavljen iz samo šestih delov, dva pri tem nista obvezna, bodo v nadaljevanju opisani vsi:

- namen – v ta del se zapiše namen modula, na primer:

```
purpose: Prikaz, kateri novorojenčki so dovzetni na okužbo  
s hepatitisom B;;
```

- razlaga – v razlagi je opisan namen in delovanje modula. Razlaga se prikaže, ko zdravstveno osebje zanima, zakaj je modul prišel do določenega zaključka:

explanation: Pri nosečnici je bil odkrit pozitiven titer površinskih antigenov hepatitisa B. Zaradi tega obstaja tveganje za razvoj hepatitisa B pri novorojenčku.;;

- ključne besede – opisne besede, po katerih se išče modul:

keywords: hepatitis B; nosečnost;;

- navedena literatura – navedba literature ni obvezna, vendar veliko pripomore k razumevanju tematike modula.; literatura mora biti oštevilčena in imeti dogovorjeno obliko; obstajata dve vrsti navedene literature, in sicer *support* literatura, ki poskrbi za razlago in s tem potrjuje rezultat, literatura navedena pod *refute* pa podaja članke, ki spodbijajo rezultat modula:

citations:

1. SUPPORT Steiner RW. Interpreting the fractional excretion of sodium. Am J Med 1984;77:699-702.

2. Goldman L, Cook EF, Brand DA, Lee TH, Rouan GW, Weisberg MC, et al. A computer protocol to predict myocardial infarction in emergency department patients with chest pain. N Engl J Med 1988;318(13):797-803.

;;

- internetne povezave – navedba internetnih povezav ni obvezna, a je uporabna. Povezave na lokalno omrežje se začnejo s predpono *file*, povezave na medmrežje pa *http*:

links:

URL "NLM Web Page", 'http://www.nlm.nih.gov/';

URL "Visible Human Project",

'http://www.nlm.nih.gov/research/visible/visible_human.html';

URL "DOS HTML File", 'file://doslinux.htm';

URL "UNIX HTML File", 'file://UnixLinux.html/';

;;

4.2.2.4 Logični del modula

V tretjem delu modula imamo podane pogoje, ki aktivirajo modul, ta pa preko logičnega računanja in sklepanja poda rezultat, ki ga posreduje zdravstvenem osebj. Zadnji del modula vsebuje:

- podatke, ki so potrebni za logična odločanja;
- prioriteto – prioriteta je podana kot število od 1 (najnižja) do 99 (najvišja) in na podlagi te številke se vršijo proženja modulov; institucije si same nastavijo vrednosti prioritete modulov tako, da ne pride do konfliktov; pri prenašanju modulov se vsem nastavi vrednost na 50;

- pogoje proženja – v tem predelu definiramo, v kakšnih pogojih se bo naš modul sprožil in začel z logičnim preračunavanjem;
- vso potrebno logiko, ki se uporabi za oceno situacije oziroma problema; na koncu logični izračun poda rezultat kot vrednot `true` ali `false`;
- akcije, ki se prožijo, če logični del vrne vrednost `true`;
- urgentnost akcije ali sporočila, ki ga vrne; urgentnost je podana kot število od 1 do 99 in se uporabi šele, ko logični del modula vrne vrednost `true`.

4.2.3 Podatkovni tipi v »Arden Syntax« jeziku

Glavne funkcije modulov MLM so pridobivanje informacij o pacientu, upravljanje z dobljenimi podatki, izdelava logične odločitve ter po potrebi izvedba z modulom povezanega postopka. Podatke za obdelavo lahko modul pridobi na različne načine, lahko neposredno iz pacientovega EZZja, lahko je konstanta v modulu samem ali pa so podatki rezultat druge operacije na drugih podatkih. Podatki so klasificirani v različne podatkovne tipe. Ti tipi so:

- `boolean` – vsebuje vrednosti `true` ali `false`; delovanje je enako kot pri Gello (tabela 1);
- `null` – vrednost `null` lahko pomeni nejasnost v pacientovih podatkih ali pa dejansko `null` vrednost v podatkovni bazi;
- število (Arden Syntax ima samo en tip števil, ki se uporabljajo, vsa računanja pa se izvajajo v plavajoči vejici);
- čas – delovanje modulov je zelo povezano s časom, saj je pomembno, kdaj se je modul zagnal, kdaj je modul sprožil določen postopek itd.;
- trajanje – podatkovni tipi trajanja izvajajo računske operacije nad podatkovnimi tipi časa in vračajo rezultat v sekundah ali za daljša obdobja v mesecih;
- znakovni nizi – znakovni nizi v Arden Syntax se morajo začeti in končati z znakom “, drugače pa niso omejeni z dolžino;
- seznam – urejeni seznam vsebuje elemente, ki so lahko ali tipa `boolean` ali dogodek, sporočilo, število, čas, trajanje, vrednost `null` ali znakovni niz; vgnezdjenih seznamov ni, kar pomeni, da elementi enega seznama ne morejo biti hkrati tudi elementi drugega; dva primera enostavnega seznama:

3, true, 5, null

- rezultati poizvedbe – pridobljeni podatki iz EZZ ali kakšne druge baze podatkov; vrnjeni rezultati se dodelijo spremenljivkam, ki jih lahko nato uporabimo.

4.2.4 Operatorji »Arden Syntax« jeziku

Operatorji se uporabljajo v izrazih za upravljanje s podatki. Operatorji uporabijo enega ali več podatkov in po obdelavi vrnejo rezultat, ki ga dodelijo na novo zgeneriranemu podatku. Sintaksa uporabe operatorjev se formira po izrazu:

$$\langle \text{num:type} \rangle := \langle \text{num:type} \rangle \text{ op } \langle \text{num:type} \rangle,$$

pri čemer je `op` operator, ki ga uporabljamo, `num` je lahko samo en element ali seznam (vendar če izvajamo operacije nad dvema seznamoma, ki nista enako dolga, nam operacija vrne vrednost `null`), `type` pa je eden od zgoraj opisanih podatkovnih tipov.

Arden Syntax uporablja ogromno število različnih operatorjev, zato bodo v nadaljevanju opisani le pogostejši, navedeni pa bodo glede na delovanje in rezultat, ki ga vrnejo. To so:

- računski operatorji (+, -, *, / in **(potenciranje));
- numerični operatorji (sin, cos, tan, exp, abs ...);
- operatorji za delo s seznamom:
 1. zlivanje (`merge`) – združi dva enako dolga seznama v nov seznam;
 2. razvrščanje (`sort`) - uredi seznam po času ali podatkih;
 3. operacija stik (,) - če uporabimo vejico med dvema seznamoma, ta seznama “prilepimo” skupaj:

$$\langle \text{n:any-type} \rangle := \langle \text{k:any-type} \rangle , \langle \text{m:any-type} \rangle$$

$$(4, "a", \text{null}) := (4, "a") , \text{null}$$

če isti operator uporabimo na samo enem elementu, se le-ta spremeni v seznam z velikostjo 1:

$$\langle \text{l:any-type} \rangle := , \langle \text{l:any-type} \rangle$$

$$(, 3) := , 3$$

- logični operatorji (`and`, `or`, `not`);

- časovni operator – vrne čas, ko je bil dobljen seznam, element ali poizvedba, itd.:

```
<n:time> := TIME [OF] <n:any-type>
```

4.2.4 Logični del modula

V logičnem delu modula se uporabljajo podatki o pacientu, testirajo se določene vrednosti in na koncu poda odločitev, ali naj se sproži določen dogodek ali ne. Logični del vsebuje vso logiko modula, ki jo uporabi za podajanje odločitev.

4.2.4.1 Spremenljivke

Spremenljivkam prirejamo vrednosti s prireditvenim stavkom:

```
<ime> := <izraz ali vrednost> ;
```

Kot primer sledi prirejanje vrednosti dveh spremenljivk `var1` in `var2`:

```
var1 := 3; //spremenljivki var1 dodelimo vrednost 3

var2 := var1 + 2;

/* spremenljivko var2 nastavimo kot seštevek var1 in števila 2. Var2 ima
torej vrednost 5.*/
```

4.2.4.2 Pogojni stavek

Pogojni stavek v Arden Syntax preveri, ali dani izraz ustreza Boolean vrednosti `true`. Če je odgovor pritrdilen, se izvedejo ukazi v notranjem bloku. Osnovni pogojni stavek je videti takole:

```
IF <izraz1> THEN //Če izraz ustreza true, se izvede blok1
    <blok1>
ENDIF;
```

Če v polje za izraz vstavimo seznam ali katero koli vrednost, ki nima vrednosti `true`, se ukazi v notranjem bloku ne izvedejo. Enostaven primer uporabe pogojnega stavka:

```
IF var1 THEN //če je spremenljivka var1 true,

var2 := 0; //je spremenljivka var2 enaka 0
    ELSE
var2 := 45; //drugače pa je var2 enaka 45
    ENDIF;
```

Na podoben način deluje tudi `while` zanka, in sicer dokler so vrednosti v izrazu enake Boolean vrednosti `true`, se blok izvaja:

```

/* Inicializacija spremenljivk */
    a_list:= ();
    m_list:= ();
    r_list:= ();
    num:= 1;

/* Preverimo vsak alergen v zdravilu, da se prepričamo, da pacient ni na
katerega alergičen */

while num <= (count med_allergen) do /*sprehodimo se čez seznam alergenov*/

allergen:= last(first num from med_allergens);
allergy_found:= (patient_allergies = allergen);
reaction:= patient_reactions where allergy_found;
medication:= med_orders where (med_allergens = allergen);

/* Alergenu, zdravilu in reakciji dodelimo spremenljivko, ki jo modul lahko
uporabi*/

If any allergy_found then          //če najdemo alergijo,
    a_list:= a_list, allergen;      //dodamo alergen v seznam alergenov

    m_list:= m_list, medication; /*dodamo zdravilo v seznam zdravil, ki jih ne
smemo dati pacientu*/

r_list:= r_list, reaction;
endif;

/* Povečamo števec za while zanko */
num:= num + 1 ;
enddo;

```

4.2.4.2 Klicanje drugih modulov

Klicanje drugih modulov je enostavno in poteka preko danih parametrov. Če kličemo modul z manj parametri, kot jih modul pričakuje, se dodatnim parametrom dodeli vrednost `null`. Pri klicanju modula s preveč parametri pa le-te klicani modul enostavno izpusti.

```

var1 := CALL klicani Modul WITH parameter1, parameter 2;

/*Če modul ne vrne nobene vrednosti, ali pa vrne vrednost null, se
spremenljivki var1 dodeli vrednost null */

```

4.2.4.3 Zaključna izjava

Z zaključno izjavo se konča logični del modula in poda zaključek. Če logični del vrne vrednost `true`, modul avtomatično prične z izvajanjem predpisanih procesov. V nasprotnem primeru pa modul svoje delovanje prekine.

```
CONCLUDE false; //podajanje zaključka
```

4.2.5 Procesni del modula

Procesni del modula se aktivira, ko logični del vrne Boolean vrednost `true`. Procesni del potem zažene procese, ki najbolj ustrezajo dani situaciji. Tipični procesi, ki se zaganjajo, so pošiljanje elektronskih sporočil osebju, dodajanje obvestil v pacientove podatke, vračanje rezultata kot modul in klicanje drugih modulov. Pravilno sprogramirani procesni del vsebuje izraze za klicanje in pisanje oziroma le vračanje določene vrednosti.

4.2.5.1 Proces pisanja

Pisanje različnih sporočil je eden od pogostejših procesov, ki jih izvajajo moduli. Proces pošlje tekst ali zakodirano sporočilo na določeno lokacijo.

```
WRITE <kodirano sporočilo>;
WRITE < kodirano sporočilo > AT <lokacija>;
      ali
WRITE <tekst>;
WRITE <tekst> AT <lokacija>;
```

Lokacije, kamor se pošiljajo sporočila, so pacientov EZZ, tiskalnik, določena baza podatkov in podani elektronski naslov.

4.2.5.2 Proces klicanja

Proces klicanja omogoča modulu MLM, da kliče druge module v odvisnosti od zaključka logičnega dela. Klicanje poteka preko imena modula in je lahko z želeno zakasnitvijo. Oblike klicanja so:

```
CALL <ime modula>; //klicanje samo po imenu
CALL <ime modula> DELAY <zakasnitev>; //z zakasnitvijo
CALL <ime modula> WITH <izraz>; //dodaten ukaz

CALL <ime modula> WITH <izraz> DELAY <zakasnitev>; //+zakasnitev

CALL <ime modula> WITH <izraz>, ..., <izraz>; //več ukazov ...
CALL <ime modula> WITH <izraz>, ..., <izraz> DELAY <zakasnitev>;
```

4.2.5.3 Proces vračanja vrednosti

Proces vračanja vrednosti se uporablja samo v modulih, ki jih kličejo drugi moduli. Izračunani rezultat v svojem logičnem delu posreduje modulu, ki ga kliče. Modul lahko vrne enega ali več rezultatov, ki jih drugi modul uporabi:

```
RETURN <rezultat>, ... , <rezultat>;
```

Proces vračanja vrednosti je zadnji proces v modulu, ki se izvede. Primer:

```
RETURN opravljene_diagnoze, imena_diagnoz;

/*modul vrača dva seznama, končane diagnoze ter seznam imen opravljenih
diagnoz*/
```

4.2.6 Modul za preverjanje alergije na penicilin s komentarjem

Kot primer modula MLM je v nadaljevanju predstavljen modul za preverjanje alergije na penicilin:

```
maintenance:
  title: Modul za preverjanje alergije na penicilin;; //naslov
  mlmname: pen_allergy;; // ime po katerem, ga lahko kličejo drugi moduli
  version: 1.00;; // verzija modula
  institution: Columbia-Presbyterian Medical Center;; //kateri instituciji
  pripada
  author: George Hripcsak, M.D.;; // avtor
  specialist: ;;
  date: 2002-03-18;; //datum izdelave
  validation: testing;; // modul je namenjen testiranju
library:
  purpose: //namen
    Ko pacientu predpišejo penicilin, preveri, če je nanj alergičen;;
  explanation: //razlaga
    Ta MLM se sproži, ko je pacientu predpisan penicilin. Če se
    ugotovi, da je nanj alergičen, se zgenerira alarmno sporočilo.;;
  keywords: penicilin; alergija;; //besede, po katerih se najde modul
  citations: ;; //navedene literature ni
knowledge:
  type: data-driven;; //tip modula
  data:
    /* naročilo penicilina je sprožilo delovanje modula */
    penicillin_order := event {medication order where
                          class = penicillin};
    /* poiščemo alergije */
    penicillin_allergy := read last {allergy where
                                   agent_class = penicillin};
    ;;
  evoke:
    penicillin_order;; //kdaj se modul aktivira (ko se predpiše
    penicilin)
logic:
  if exist(penicillin_allergy)then //če alergija obstaja, modul vrne true
    conclude true;
  endif;
  ;;
  action: //če logični del vrne true, modul pošlje sporočilo
    write"Pozor, pacient ima zabeleženo alergijo na penicilin:"||
    penicillin_allergy;;
    //na koncu sporočila se še doda (||)dobljena vrednost iz baze
    //(penicillin_allergy)
  urgency: 50;; //stopnja pomembnosti sporočila
end: //konec modula
```

5. Zaključek

Namen diplomske naloge je bil spoznati projekt eZdravje, predstaviti del njegovih standardov ter bližje spoznati organizacijo HL7, ki postavlja podatkovne standarde v zdravstvu. Projekt eZdravje velja za enega največjih projektov informatizacije javnih storitev v Sloveniji, ki pa je z razvojem še daleč od zastavljenega cilja. Projekt je ministrstvo za zdravje lansiralo septembra leta 2008 in je vseboval skupno več kot 20 podprojektov. Do maja 2010 (24% zastavljenega časa) sta bila zaključena samo dva podprojekta (9% celotnega projekta), in sicer študija izvedljivosti eZdravja ter informacijska varnost prihajajočega elektronskega zdravstvenega omrežja (zNET). Torej bo do aktivne uporabe v diplomski nalogi opisanih standardov preteklo še kar nekaj časa.

Organizacija HL7 se sicer razglša za neprofitno organizacijo, vendar če želiš več informacij o njihovih standardih, kot jih ponujajo na spletu, je to možno le preko njihovih plačljivih tečajev ali preko plačljivih dokumentov. Člani organizacije imajo prost dostop do njihovih standardov in določene dokumentacije, vendar znaša letna članarina od 140 dolarjev (omejeno članstvo brez dostopa do standardov, namenjeno študentom v ZDA) do 37.400 dolarjev (sponzorji in podporniki organizacije). Ko sem ob pisanju diplomske naloge analiziral njihove standardizirane povpraševalni jezik Gello, sem imel veliko težav pri iskanju informacij o jeziku samem in o njegovi uporabi. Organizacijo HL7 sem večkrat skušal kontaktirati za kakšne dodatne informacije o jeziku, vendar mi niso niti odgovorili. Zanimivo je tudi to, da Slovenija, kljub temu da je članica organizacije HL7, kot taka ni navedena na njeni uradni strani. Možno je, da zaradi tega, ker je projekt eZdravje šele v svoji začetni fazi in sodelovanje z organizacijo HL7 še ni steklo.

Standardni povpraševalni jezik Gello se bo uporabljal po vzpostavitvi konsistentnega podatkovnega modela v eZdravju. Njegove prednosti pred povpraševalnim jezikom SQL ter dejstvo, da član organizacije HL7 dobi dostop do vseh standardov, bosta zagotovila njegovo uporabo tudi pri nas.

Moduli, kot so Medical Logic Modules, pa po mojem mnenju predstavljajo prihodnost sodobne medicine. Ni nujno, da ravno ti, ki uporabljajo standardizirani jezik Arden Syntax; jih pa unikaten način delovanja ter mobilnost postavljata v ospredje informatike v medicini. Ker smo ljudje zmotljivi in površni je prisotnost kontrolnega sistema, ki vsako odločitev dodatno oceni, logično sklepa ter poda svoje mnenje, zelo dobrodošla. Ti moduli se avtomatično prožijo in skrbijo za pravilno odločanje in informiranje zdravstvenega osebja.

6. Viri

- [1] Projekt eZdravje. Zadnji dostop: 5. avgust 2010. Dostopno na: <http://www.mz.gov.si/si/ezdravje/>
- [2] Nacionalna uvedba sodobnega ON-LINE elektronskega poslovanja v sistem zdravstvenega varstva in zdravstvenega zavarovanja. Zadnji dostop: 5. avgust 2010. Dostopno na: <http://www.zzzs.si/zzzs/internet/zzzs.nsf/o/2B02EFC719EABA9AC125737C0038FA1C?OpenDocument>
- [3] Kratka predstavitev projekta eZdravje. Zadnji dostop: 5. avgust 2010. Dostopno na: http://www.mz.gov.si/fileadmin/mz.gov.si/pageuploads/eZdravje/predstavitev/N_Kratka_predstavitev_projekta_za_splet_feb10.pdf
- [4] e-Zdravje²⁰¹⁰ Strategija informatizacije slovenskega zdravstvenega sistema 2005–2010. Dostopno na: http://www.mz.gov.si/si/delovna_podrocja/zdravstveno_varstvo/sluzba_za_informatiko/informatika_v_zdravstvu_siz_ozis/eZdravje116slo.doc
- [5] Standard openEHR. Zadnji dostop: 5. avgust 2010. Dostopno na: <http://en.wikipedia.org/wiki/OpenEHR>
- [6] Health Level Seven International (HL7). Zadnji dostop: 5. avgust 2010. Dostopno na: <http://www.hl7.org/>
- [7] Referenčni model OSI. Zadnji dostop: 5. avgust 2010. Dostopno na: http://sl.wikipedia.org/wiki/ISO/OSI_referen%C4%8Dni_model
- [8] International Organization for Standardization (ISO). Zadnji dostop: 5. avgust 2010. Dostopno na: <http://www.iso.org/iso/about.htm>
- [9] Ministrstvo za zdravje postalo organizacijski član Health Level Seven. Zadnji dostop: 5. avgust 2010. Dostopno na: <http://www.mz.gov.si/si/ezdravje/novice/PType/98/>
- [10] Versions of the HL7 Standard. Zadnji dostop: 5. avgust 2010. Dostopno na: <http://www.corepointhealth.com/resource-center/hl7-resources/hl7-standard-versions>
- [11] OpenClinical – knowledge management for medical care. Zadnji dostop: 5. avgust 2010. Dostopno na: <http://www.openclinical.org/home.html>
- [12] GELLO a decision support language. Zadnji dostop: 5. avgust 2010. Dostopno na: <http://ehealthaustralia.org/gello-a-decision-support-language/>
- [13] GELLO: A Common Expression Language. Zadnji dostop: 5. avgust 2010. Dostopno na: http://aps1.medical-objects.com.au/gello/gello_v1_1.htm

-
- [14] The Biomedical and Health Informatics (BHI) Scholar Blog Arden Syntax. Zadnji dostop: 5. september 2010. Dostopno na: <http://www.pitt.edu/~rdb20/bhi-scholar-blog/Arden%20Syntax.psp.html>
- [15] Petros Kyprianides 810:188G Arden Syntax. Zadnji dostop: 15. september 2010. Dostopno na: http://neamh.cns.uni.edu/MedInfo/arden_syntax.html
- [16] HL7 Arden Syntax. Zadnji dostop: 15. september 2010. Dostopno na: <http://www.hl7.org/library/Committees/Arden/Arden.PDF>