

Univerza v Ljubljani
Fakulteta za računalništvo in informatiko
Fakulteta za matematiko in fiziko

Matej Pucihar

**Analiza implementacije eNaročanja v
bolnišničnih informacijskih sistemih**

DIPLOMSKO DELO
INTERDISCIPLINARNI UNIVERZITETNI
ŠTUDIJSKI PROGRAM PRVE STOPNJE
RAČUNALNIŠTVO IN MATEMATIKA

prof. dr. Miha Mraz
MENTOR

Ljubljana, 2019

© 2019, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Rezultati diplomskega dela so intelektualna lastnina Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje Fakultete za računalništvo in informatiko ter mentorja.

Univerza
v Ljubljani

Fakulteta za računalništvo
in informatiko



Tematika naloge:

Kandidat naj v svojem delu predstavi problematiko elektronskega naročanja na zdravstvene storitve v Sloveniji gledano z nacionalnega vidika. Izvede naj analizo sistema eNaročanja preko eNapotnic, njegovo vpetost v ostale zdravstvene informacijske sisteme kot so npr. bolnišnični informacijski sistemi in nacionalni centralni register pacientovih podatkov ter izpostavi njegove slabosti. Za predstavljene slabosti naj predlaga tudi možne izboljšave.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani izjavljam, da sem avtor dela, da slednje ne vsebuje materiala, ki bi ga kdorkoli predhodno že objavil ali oddal v obravnavo za pridobitev naziva na univerzi ali drugem visokošolskem zavodu, razen v primerih kjer so navedeni viri.

S svojim podpisom zagotavljam, da:

- sem delo izdelal samostojno pod mentorstvom prof. dr. Mihe Mraza,
- so elektronska oblika dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko in
- soglašam z javno objavo elektronske oblike dela v zbirki "Dela FRI".

— Matej Pucihar, Ljubljana, marec 2019.

Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Matej Pucihar
Analiza implementacije eNaročanja v bolnišničnih informacijskih sistemih

POVZETEK

Diplomska naloga prikaže slovenski sistem informacijske rešitve elektronskega naročanja na zdravstvene storitve tako, da skozi oči izvajalca zdravstvenih storitev prikaže primere uporabe elektronske napotnice in tako predstavi poslovno logiko eNaročanja. Obrazložena je arhitektura izvedbe rešitve eNaročanja, kjer je prikazana komunikacija med akterji udeleženi v eNaročanju. Izvajalci zdravstvenih storitev shranjujejo napotnice v centralni register podatkov o pacientu in izvajajo naročila. Pri tem za poslovno logiko kot vmesnik med registrom in izvajalci nastopa centralni sistem naročanja, ki hkrati omogoča naročanje preko spleta na portalu zVEM vsakemu posamezniku, kateremu je bila izdana elektronska napotnica. Za avtentikacijo in avtorizacijo tako kot za vse ostale projekte eZdravja skrbi varnostna shema.

Prikazana je tehnična izvedba vzpostavitve eNaročanja v bolnišničnem informacijskem sistemu Think!Clinical z uporabo različnih metod, ki sklajujejo podatke o napotnicah in naročilih bolnišničnega informacijskega sistema z stanjem podatkov centralnega registra podatkov o pacientu. Pri tem so izpostavljene številne pomanjkljivosti eNaročanja v prvotni izvedbi.

Opisali smo vzorčna primera nadgradnje integracijskih metod na novo verzijo specifikacije eNaročanja, ki odpravlja večino vsebinskih težav sistema elektronskega napotovanja. Kot največja obstoječa pomanjkljivost izvedbe eNaročanja je izpostavljeno in utemeljeno dejstvo, da je potrebno za večkratne napotnice podpreti kreacijo večih aktualnih naročil hkrati.

Potencialna rešitev je opredeljena kot poenotenje poslovne logike eNaročanja s konkretnimi scenariji naročanja v bolnišnicah, kar pomeni, da je potrebno omogočiti kreacijo

naročil brez napotnice in obstoj večih aktualnih naročil za isto VEČKRATNO napotnico hkrati. To je cilj, h katerem stremimo tako razvijalci kot tudi uporabniki sistema eNaročanja.

Ključne besede: elektronsko naročanje, elektronska napotnica, centralni sistem naročanja, hospitalni sistem naročanja, centralni register podatkov o pacientih, varnostna shema, vrsta zdravstvene storitve

University of Ljubljana
Faculty of Computer and Information Science

Matej Pucihar

Analysis of electronic ordering implementation in hospital information systems

ABSTRACT

The thesis presents the Slovenian electronic solution for ordering to medical services by showing examples of use of a digital referral through the eyes of the health service provider and presenting the business logic of electronic ordering. The architecture of the implementation of the electronic ordering solution is explained, where the communication between the systems involved in electronic ordering is shown. Health care providers store referrals in the central register of patient data and perform orders. In this regard central ordering system acts as the interface between the data registry and the performers. It also enables online ordering on the portal zVEM to every individual to whom an electronic referral has been issued. As for all other Slovenian eHealth projects, a security scheme system is used for authentication and authorization.

The technical implementation of the setting up of electronic ordering in the hospital information system Think!Clinical is demonstrated using a variety of methods that coordinate data of referrals and orders of hospital information systems with the state of the data stored by the central patient data register. A number of shortcomings in the initial implementation of the electronic ordering implementation are also highlighted.

We have described a sample case of upgrading integration methods to a new version of the eOrdering specification that eliminates most of the content problems of the electronic ordering system. As the biggest existing shortcoming of the current implementation of electronic ordering, the fact is that it is necessary to support the creation of several current orders at the same time for referrals, which are valid through period of time is highlighted. This is the goal that both developers and users of the eOrdering system are aiming for.

Key words: electronic ordering, digital referral, central ordering system, hospital ordering system, central registry of patient data, security scheme, medical procedure type

ZAHVALA

Zahvaljujem se mentorju za nasvete, znanje, in pomoč pri iskanju konceptualnih rešitev obstoječih težav sistema eNaročanja. Zahvaljujem se tudi prijateljem, družini, Marjeti Boštjančič in sodelavcem podjetja Marand d.o.o., ki so mi pomagali pri razvoju in nadgradnji eNaročanja v sistemu Think!Clinical.

— Matej Pucihar, Ljubljana, marec 2019.

KAZALO

Povzetek	i
Abstract	iii
Zahvala	v
1 Uvod	1
1.1 Motivacija	1
1.2 Cilj diplomske naloge	2
1.3 Struktura	2
2 Opis problema	5
2.1 Napotnica	5
2.2 Centralni register podatkov o pacientih	7
2.3 Varnostna shema eZdravje	7
2.4 Think!Clinical	7
2.5 Centralni sistem naročanja	8
2.6 Portal zVEM	9
2.7 Shema gradnikov eNaročanja	9
2.8 Življenjski cikel elektronske napotnice	11
3 Vzpostavitev eNaročanja iz HOS sistema	17
3.1 Opis delovanja eNaročanja pri najpogostejših scenarijih	17
3.1.1 Izdaja napotnice preko portala zVem in kreacija naročila iz HOS sistema	18
3.1.2 Izdaja napotnice iz HOS sistema in kreacija naročila preko portala zVEM	19

3.1.3	Izdaja napotnice, kreacija in preklic naročila preko portala zVEM	21
3.2	Opis metod integracije med COS in HOS sistemi	22
3.2.1	HOS - COS	23
3.2.2	COS - HOS	25
3.3	Pomanjkljivosti sistema eNaročanje V1	26
3.3.1	Težave uporabnikov HOS sistemov	26
3.3.2	Težave HOS sistemov pri implementaciji integracije	28
4	Izboljšave sistema eNaročanja	31
4.1	Izboljšave eNaročanja V2	31
4.2	Vzorčna primera izvedbe nadgradnje na novo verzijo	33
4.2.1	Nadgradnja funkcije <code>getAppointmentForProcedure()</code>	34
4.2.2	Nadgradnja funkcije <code>cancelReferral()</code>	38
4.3	Način prehoda	41
5	Zaključek	45

1 Uvod

Pričujoče delo obravnava nacionalno rešitev elektronskega naročanja na zdravstvene storitve.

1.1 Motivacija

V sodelovanju s podjetjem Marand d.o.o. sem se kot del ekipe, ki razvija in vzdržuje projekt informacijskega sistema Think!Clinical, dobro spoznal tako z administrativno platjo vsakodnevnih opravil medicinskega osebja v slovenskih bolnišnicah, kot tudi s samimi informacijskimi tehnologijami, ki jih uporablja sodoben informacijski sistem. Posledično sem se spoznal tudi z integracijami sistema z različnimi zunanjimi sistemi, med katere spada tudi večina storitev projekta eZdravja, ki ga vodi Nacionalni inštitut za javno zdravje (NIJZ). eNaročanje je vsekakor najkompleksnejša in najbolj razširjena storitev projekta eZdravje.

Za tematiko moje diplomske naloge sem se odločil na podlagi sledečih dejstev:

- kvaliteta in zanesljivost storitev eNaročanja temelji na zanesljivi in uspešni integraciji informacijskih sistemov izvajalcev zdravstvenih dejavnosti s centralnim sis-

temom eNaročanja;

- integracija informacijskih sistemov izvajalcev zdravstvenih dejavnosti z eNaročanjem vse od svoje uveljavitve leta 2015 izmed vseh integracij povzroča daleč največ težav tako pri samem naročanju in izvedbi naročil, kot tudi pri sami implementaciji, vzdrževanju in nadgradnji poslovne logike integracije sistema z eNaročanjem;
- eNaročanje je nacionalni projekt, katerega neuspehe financiramo davkoplačevalci.

1.2 Cilj diplomske naloge

eNaročanje je projekt eZdravja, ki je bil realiziran v želji po centralizaciji in digitalizaciji naročanja na zdravstvene storitve sekundarne in terciarne zdravstvene ravni. Običajni državljani tako pridobimo možnost prostega naročanja preko spleta, medtem ko država prepreči možnost prirejanja podatkov zdravstvenih ustanov in tako uvede nadzor s pomočjo zanesljivih podatkov tako o čakalnih vrstah, kot tudi o sami učinkovitosti izvajalcev zdravstvenih dejavnosti.

Cilj diplomske naloge je skozi analizo informacijske rešitve eNaročanja predstaviti sledeče tematike:

- funkcionalnosti in gradnike sistema eNaročanja;
- implementacijo integracije informacijskega sistema izvajalca zdravstvenih dejavnosti s centralnim sistemom eNaročanja in njegove pomanjkljivosti;
- težave, ki so nastale na podlagi uvedbe eNaročanja;
- izboljšave in implementacijo eNaročanja verzije 2;
- obstoječe težave v verziji 2; konceptualna rešitev bo predstavljena v zaključku diplomskega dela.

1.3 Struktura

V poglavju 2 je opisana domena problema eNaročanja. Problem je predstavljen skozi opise glavnih terminov, gradnikov in povezav med gradniki sistema eNaročanja. V poglavju 3 v prvem razdelku sledi konkretna predstavitev delovanja poslovne logike eNaročanja skozi nekaj primerov uporabe sistema. V drugem razdelku poglavja 3 so

definirane in predstavljene metode, preko katerih centralni sistem eNaročanja komunicira z izvajalci zdravstvenih dejavnosti oziroma njihovimi informacijskimi sistemi, v zadnjem oziroma tretjem razdelku poglavja 3 pa so skozi oči uporabnika in razvijalca informacijskega sistema opisane težave, ki so nastale kot posledica uvedbe eNaročanja. V poglavju 4 je opisana nadgradnja eNaročanja na verzijo 2. V prvem razdelku poglavja 4 so opisane rešitve, ki jih prinaša nova verzija za izvajalce zdravstvenih dejavnosti. V drugem razdelku poglavja 4 je na kratko opisana tehnologija, s pomočjo katere poteka komunikacija med sistemi, nato pa sta opisana dva vzorčna primera nadgradnje funkcije na novo verzijo. V zadnjem oziroma tretjem razdelku je opisan način prehoda med verzijama z vidika razvijalca bolnišničnega informacijskega sistema.

2 Opis problema

eNaročanje predstavlja elektronski sistem napotitve pacientov s primarne na sekundarno ali terciarno raven ali znotraj sekundarne ali terciarne ravni zdravljenja. Pacientom prinaša prosto izbiro zdravstvene ustanove glede na ponujene termine ustanov. Tako je mogoča izbira optimalnega termina. Podprto je tudi obveščanje pacientov o točnem terminu obiska in morebitnih prenaročilih. Za vsako zdravstveno ustanovo je možen tudi vpogled v njene čakalne vrste (okvirni termin) in čakalne knjige (naročilo na urniku, točen termin) za vsako izmed zdravstvenih storitev, ki jih ustanova izvaja, ter pregled realiziranih in nerealiziranih naročil ter izkoriščenih in neizkoriščenih napotnic. V sledečih razdelkih sledijo opisi glavnih tērminov in gradnikov sistema eNaročanja.

2.1 Napotnica

Napotnica je listina, s katero izvajalci (izbrani osebni zdravniki in napotni zdravniki z ustreznim pooblastilom) napotijo zavarovano osebo na zdravstvene storitve specialistične ambulantne ali bolnišnične zdravstvene dejavnosti. Z napotnico izbrani osebni zdravnik ali po njegovem pooblastilu napotni zdravnik prenaša pooblastila na druge zdravnike v

specialistično-ambulantni ali bolnišnični zdravstveni dejavnosti [1].

V Sloveniji imamo dva tipa napotnice, in sicer papirno in elektronsko. Od 10. aprila 2017 sta omenjeni napotnici izenačeni, kar pomeni, da z vpeljavo elektronske napotnice (eNapotnice) bolnikom ni potrebno pošiljati papirnatih zelenih obrazcev za naročilo k zdravniku specialistu. Papirna napotnica vsebuje naslednje podatke:

- **Nujnost:** REDNO, HITRO, ZELO HITRO, NUJNO; na podlagi nujnosti je definirana maksimalna dopustna čakalna doba;
- **Veljavnost napotnice:** ENKRATNA, VEČKRATNA in število mesecev veljavnosti napotnice;
- **Administrativni podatki napotnega zdravnika in ustanove;**
- **Administrativni podatki pacienta:** ime, priimek, kontakt, starost, ZZZS številka itd.;
- **RDP:** radiološka preiskava;
- **Prednostni kriteriji:** dojenje, nosečnost, ocena nezmožnosti za delo itd.;
- **Razlog obravnave:** bolezen, poškodba izven dela, poklicna bolezen itd.;
- **Obseg pooblastila,** ki ga zdravnik, ki izdaja napotnico, prenaša na napotnega zdravnika;
- **Dodatni medicinski podatki.**

Elektronska napotnica vsebuje vse podatke papirne napotnice, dodanih pa je nekaj administrativnih podatkov, potrebnih za delovanje poslovne logike eNaročanja. Ti podatki so sledeči:

- **VZS:** vrsta zdravstvene storitve, na katero je pacient napoten s strani zdravnika napotovalca;
- **Status:** IZDANA, VPISANA, V UPORABI, IZKORIŠČENA, NI IZKORIŠČENA in PREKLICANA;
- **Interval veljavnosti napotnice,** ki je na podlagi veljavnosti določen ob prvem sprejemu za VEČKRATNE napotnice.

2.2 Centralni register podatkov o pacientih

Centralni register podatkov o pacientih (CRPP) je zbirka elektronskih zdravstvenih zapisov, ki predstavlja osnovo za celostno in kontinuirano obravnavo pacienta v zdravstvenem sistemu. CRPP je sistem, ki omogoča delovanje vseh storitev eZdravja. Zbirka vsebuje povzetek osnovnih podatkov o pacientu in z zakonodajo določeno pacientovo zdravstveno dokumentacijo. CRPP omogoča zdravstvenemu osebju, ki sodeluje pri zdravstveni oskrbi pacienta, vpogled v ključne podatke, s katerimi lahko zagotavlja primerno, varno in kakovostno oskrbo. Zdravstvenim delavcem tako v Sloveniji kot v tujini omogoči nemoteno komunikacijo, varno in sledljivo izmenjavo podatkov ter zanesljivo ažuriranje podatkov [2].

Ob zdravstveni obravnavi se v CRPP vnašajo zdravstveni podatki, kot so npr. cepljenja in alergije. Shranjuje se tudi zdravstvena dokumentacija, kot npr. ambulantni izvidi, odpustna pisma, eRecepti in eNapotnice.

CRPP deluje nad Marandovo Think!EHR platformo [3], kar pomeni da so vsi podatki shranjeni v strukturirani elektronski zdravstveni obliki, ki temelji na neodvisnih arhetipih openEHR specifikacije [4].

2.3 Varnostna shema eZdravje

Varnostna shema eZdravje (VS) je aplikacija, ki omogoča enotno upravljanje z uporabniki in njihovimi pravicami za delo v različnih aplikacijah. Izvaja avtentikacijo in avtorizacijo uporabnikov. Podpira upravljanje s pravicami sistemskih uporabnikov ali aplikacij [5].

VS uporablja SAML, CAS in XACML protokole, omogoča pa SSO (angl. *single sign-on*), kar pomeni, da uporabnik ob prijavi v VS pridobi "SAML security token" oziroma žeton varnostne sheme, ki nato služi za avtentikacijo in avtorizacijo uporabnika v vseh ostalih sistemih eZdravja (eNaročanje, eRecept, CRPP itd.).

VS ne ponuja standardnega LDAP vmesnika, ki bi zagotavljal usklajevanje uporabnikov in njihovih pravic v sistemih, ki uporabljajo VS.

2.4 Think!Clinical

Think!Clinical je klinični informacijski sistem, ki je v uporabi na Onkološkem inštitutu Ljubljana in na nekaterih klinikah Univerzitetnega kliničnega centra Ljubljana (npr. na Kliniki za nuklearno medicino, Pediatrični kliniki, Kliniki za infektivne bolezni in

vročinska stanja itd.).

Think!Clinical je v sistem eNaročanja integriran kot eden izmed bolnišničnih sistemov naročanja (angl. *hospital ordering system* - HOS).

Sistem je zgrajen na klasični server-client infrastrukturi. To pomeni, da je večina poslovne logike aplikacije implementirane na strežniku. Strežnik praviloma skrbi tudi za integracijo z zunanjimi sistemi, med katerimi je tudi eNaročanje. Izjema je VS, v katero se uporabnik prijavi z digitalnim potrdilom, ki ga ima na svoji pametni kartici. Privatni ključ certifikata pametne kartice je neizvozen (angl. *unexportable*), dostopen pa je samo preko programskih knjižnic proizvajalca čitalnikov pametnih kartic. Integracijo z VS je torej v opisani obliki nemogoče implementirati na strežniku.

2.5 Centralni sistem naročanja

Centralni sistem naročanja (angl. *central ordering system* - COS) je sistem, na katerem je implementirana poslovna logika eNaročanja. Sistem pozna dva ključna koncepta, in sicer *napotnico* in *naročilo*. Podatke napotnice lahko ločimo na dva dela, in sicer na medicinske podatke ter administrativne podatke. Medicinske podatke napotnice COS hrani v CRPP, administrativne podatke napotnice in podatke o naročilih pa hrani COS. Administrativni podatki so z vidika poslovne logike naročanja najbolj zanimivi podatki, zato se na medicinske podatke ne bomo osredotočali.

Med administrativne podatke napotnice uvrščamo sledeče podatke:

- **identifikator napotnice:** 13-mestno število, ki ga generira COS;
- **vrsta zdravstvene storitve (VZS):** definirana s strani zdravnika napotovalca;
- **interval veljavnosti napotnice:** določen s strani centralnega sistema ob prvem sprejemu za VEČKRATNO napotnico;
- **status napotnice:** v ciklu naročila se spreminja;

Naročilo vsebuje naslednje podatke:

- **identifikator naročila:** generiran s strani HOS sistema, kamor je pacient naročen;
- **podatki ustanove;**
- **ime, priimek, EMŠO in ZZS pacienta;**

- **VZS;**
- **identifikator napotnice;**
- **čas predvidenega obiska:** okvirni (čakalna vrsta) ali točen termin;

2.6 Portal zVEM

Portal zVEM (zdravjeVsenaEnemMestu) omogoča varen dostop do storitev eZdravja, z možnostjo eNaročanja in dostopa do eReceptov ter elektronskih zdravstvenih podatkov [6].

Portal pacientom omogoča prijavo z ali brez kvalificiranega digitalnega potrdila. Če se pacient prijavi brez digitalnega potrdila mora navesti svojo ZZZS številko in identifikator eNapotnice. V portalu ima možnost pregleda eNapotnice in okvirnih terminov vseh izvajalcev VZS določene na napotnici in eNaročanja. Preko portala se pacient lahko tudi naroči na obravnavo in prav tako lahko naročilo prekliče.

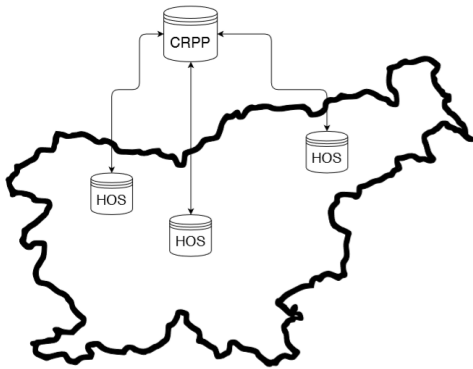
Medicinsko osebje se v portal zVEM prijavi s svojim digitalnim potrdilom preko VS. Glede na pravice določene v VS imajo možnost eNaročanja, pregled svojih izdanih napotnic, izdaje nove napotnice itd.

2.7 Shema gradnikov eNaročanja

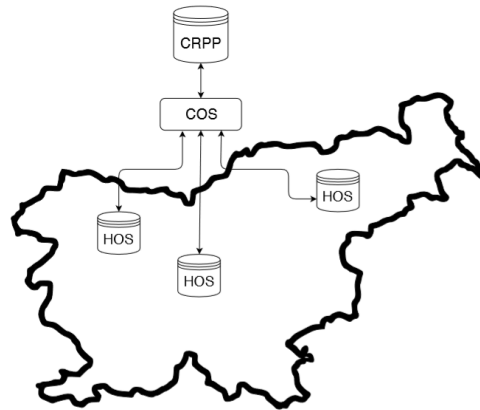
Osnovna ideja je, da informacijski sistemi zdravstvenih ustanov (HOS) izdajo napotnico v elektronski obliki. Napotnica se shrani v CRPP. Shema takega sistema se nahaja na sliki 2.1a. Na podlagi napotnice je potrebno kreirati naročilo. Naročila in njihova poslovna logika ne morata biti implementirana v CRPP, zato implementiramo centralni sistem naročanja (COS), ki bo skrbel za poslovno logiko naročanja med HOS sistemi in CRPP bazo pacientovih podatkov. Shema takega sistema se nahaja na sliki 2.1b.

Potrebujemo še sistem, prek katerega je možno naročati tudi izven HOS informacijskih sistemov, ker potrebujemo tudi možnost izdaje napotnic za tiste zdravnike, ki jim lokalni HOS informacijski sistem tega ne omogoča. Na shemo dodamo portal zVEM. Shema takega sistema se nahaja na sliki 2.1c.

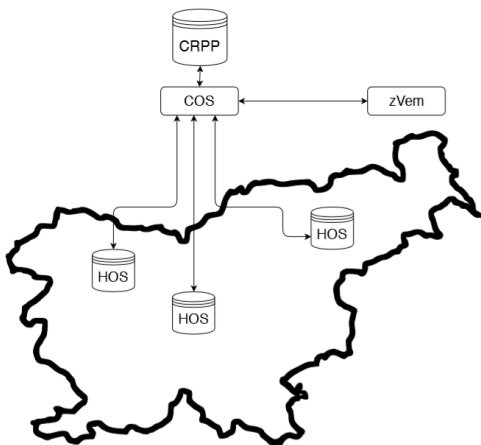
Manjka nam samo še sistem, iz katerega pridobimo informacijo, ali je uporabnik avtoriziran za izvedbo določene akcije v COS. Na shemo dodamo VS. Shema celotnega sistema se nahaja na sliki 2.1d.



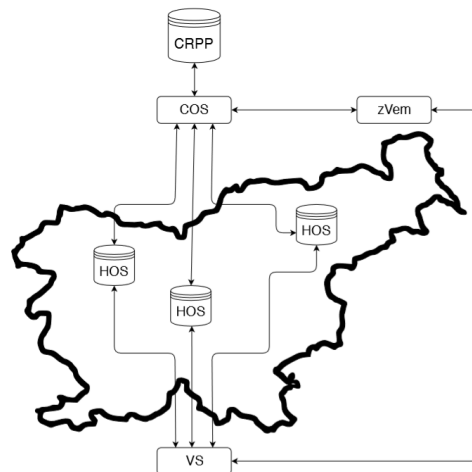
(a) HOS sistemi shranjujejo napatnice v CRPP.



(b) Shema, kjer je za potrebe poslovne logike naročanja dodan COS.



(c) Shema z dodanim portalom zVEM.



(d) Na shemo dodana še VS. Celotna shema sistema eNaročanja.

Slika 2.1: Sheme gradnikov sistema eNaročanja. Na osnovno shemo 2.1a smo postopoma dodajali gradnike sistema tako, da smo v 2.1d prišli do celotne sheme sistema eNaročanja.

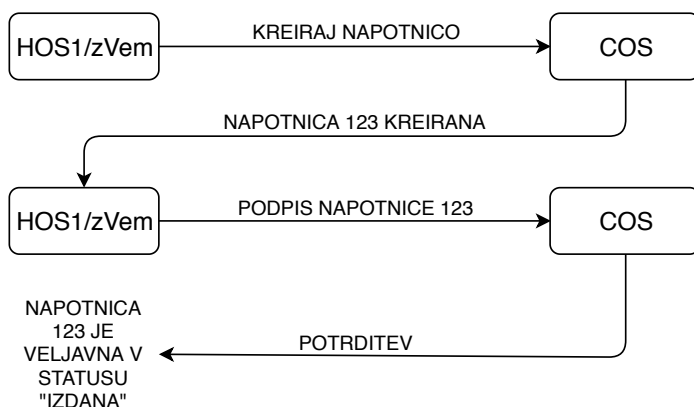
2.8 Življenjski cikel elektronske napotnice

Pred naslednjim poglavjem je potrebno še vsebinsko razložiti stanja napotnice in naročila. Naročilo ne pozna stanja. Stanje, v katerem je naročilo, se odraža na stanju oziroma **statusu** napotnice. Nabor možnih vrednosti statusov napotnice je sledeč:

- **IZDANA** – status, ki ga eNapotnica dobi v trenutku ustvarjanja oz. vnosa v sistem s strani zdravnika primarnega zdravstvenega varstva ali zdravnika specialista oz. njune medicinske sestre;
- **VPISANA** – status, ki ga eNapotnica dobi v trenutku, ko pacient dobi termin ali je vpisan v interno čakalno vrsto v zdravstveni ustanovi na podlagi napotnice;
- **V UPORABI** – eNapotnica dobi ta status v trenutku sprejema pacienta v zdravstveno ustanovo;
- **IZKORIŠČENA** – eNapotnica dobi ta status v trenutku realizacije naročila (ob izdelavi prvega izvida na ambulantni ali odpustnice na bolnišničnem oddelku);
- **NEIZKORIŠČENA** – eNapotnica dobi ta status, če pacient v obdobju, ki ga bo lahko določil ZZS (trenutno je obdobje neomejeno), ne opravi nobene storitve za to napotnico;
- **PREKLICANA** – eNapotnica dobi ta status v trenutku razveljavitve (stornacije);

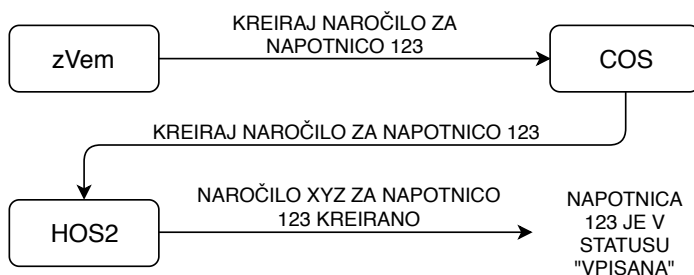
Tipična napotitev s primarnega na sekundarni ali terciarni nivo zdravstva poteka v sledečih korakih:

- Osebni zdravnik izda in podpiše napotnico. Napotnica dobi status IZDANA, kar pomeni, da v tem trenutku ne obstaja aktualno naročilo za to napotnico. Zdravnik to lahko naredi na portalu zVEM ali pa na katerem izmed zalednih HOS sistemov. Shema opisanega podatkovnega toka se nahaja na sliki 2.2.

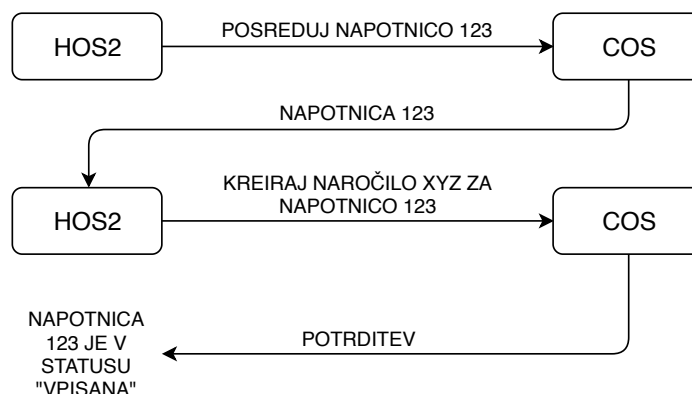


Slika 2.2: Integracijski podatkovni tok med izdajo in podpisom napotnice.

- V nekem HOS sistemu se kreira naročilo oziroma vpis v čakalno vrsto za predhodno izdano eNapotnico. Naročilo je definirano na urniku, torej poznamo točen termin obiska. Vpis v čakalno vrsto pozna samo okvirni termin. eNapotnica v tem trenutku zamenja status iz IZDANA v VPISANA. Naročilo je kreirano samo za napotnice s statusom IZDANA. Kreiranje naročila za napotnico tako obravnavamo tudi kot zaklepanje napotnice drugim ustanovam. Naročilo v HOS sistemu je lahko kreirano preko portala zVEM ali preko HOS sistema, kjer administracija sprejme napotnico. Shemi opisanih podatkovnih tokov za oba primera se nahajata na sliki 2.3 in 2.4.

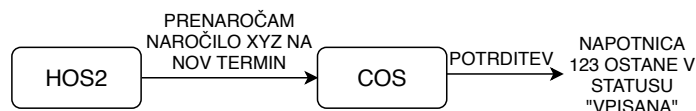


Slika 2.3: Integracijski podatkovni tok pri kreaciji naročila iz HOS sistema.



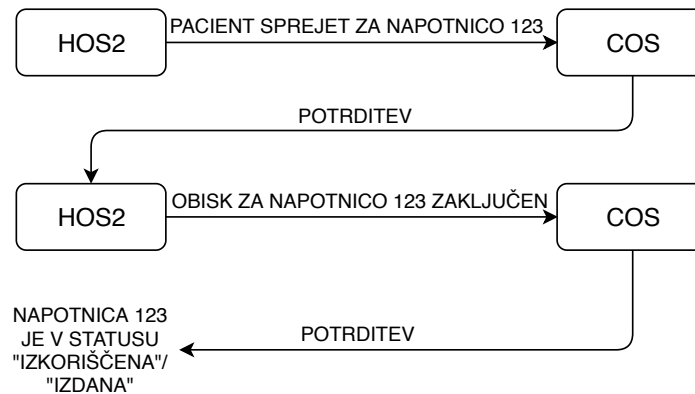
Slika 2.4: Integracijski podatkovni tok pri kreaciji naročila iz portala zVem.

- Predpostavimo, da je bila ob sprejemu napotnica uvrščena v čakalno vrsto. Ko napotnica pride do prvega mesta čakalne vrste, jo administracija uvrsti na urnik. Takrat je kreirano dejansko naročilo na urniku, v COS pa se pošljejo podatki o prenaročanju iz okvirnega na točno določen termin. Pacientu je poslano vabilo za pregled. Napotnica ostane v statusu VPISANA. Shema opisanega podatkovnega toka se nahaja na sliki 2.5.



Slika 2.5: Integracijski podatkovni tok pri uvrstitvi napotnice na urnik oziroma prenaročanje na točen termin.

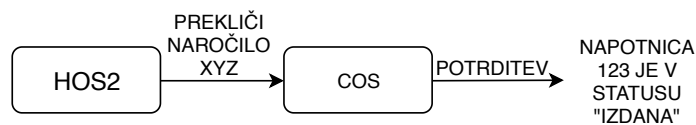
- Pacient pride na obravnavo. Ob sprejemu v ambulanto se napotnici spremeni status iz VPISANA v V UPORABI. Ko je obravnava zaključena se enkratni napotnici status spremeni v IZKORIŠČENA, večkratni pa se status spremeni v IZDANA. Določen je interval veljavnosti napotnice. Shema opisanega podatkovnega toka se nahaja na shemi 2.6.



Slika 2.6: Integracijski podatkovni tok pri sprejemu in realizaciji naročila.

- Predhodno smo opisali najpogostejši scenarij, ko je brez zapletov kreirano naročilo, sprejem in obisk. Seveda pa ima pacient možnost naročilo tudi preklicati. Pacient to lahko naredi preko portala zVEM ali pa kontaktira kliniko, kjer je nato naročilo preklicano. Shemi opisanih podatkovnih tokov za oba primera se nahajata na shemi 2.7 in 2.8.

Tukaj velja omeniti tudi dejstvo, da naročila za napotnico v statusu V UPORABI ne moremo direktno preklicati. Najprej je potrebno preklicati pacientov sprejem, nato lahko šele prekličemo samo naročilo. Podoben scenarij imamo tudi pri izbrisu obiska. Potrebno je preklicati obisk, sprejem in naročilo.



Slika 2.7: Integracijski podatkovni tok med preklicom naročila iz HOS sistema.



Slika 2.8: Integracijski podatkovni tok med preklicom naročila iz portala zVEM.

3 Vzpostavitev eNaročanja iz HOS sistema

V predhodnem poglavju smo prikazali, da je eNaročanje sestavljeno iz več komponent. V poglavju vzpostavitev eNaročanja iz HOS sistema bomo v razdelku 3.1 opisali in prikazali interakcijo med sistemi eNaročanja ob nekaj najpogostejših scenarijih uporabe napotnice, v razdelku 3.2 pa sledi opis metod integracije med COS in HOS sistemi. V razdelku 3.3 sledijo opisi problemov, ki jih je povzročila implementacija integracije eNaročanja verzije 1 v HOS sisteme.

3.1 Opis delovanja eNaročanja pri najpogostejših scenarijih

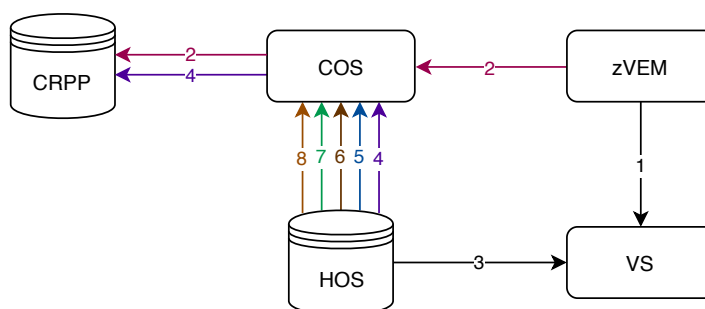
V naslednjih podrazdelkih sledijo opisi nekaj najpogostejših scenarijev uporabe napotnice. Opisi poleg opisa interakcije med sistemi eNaročanja vključujejo tudi okvirni opis interne poslovne logike HOS sistema in uporabniške izkušnje.

3.1.1 Izdaja napotnice preko portala zVem in kreacija naročila iz HOS sistema

V tem razdelku opišemo scenarij, kjer pacientu zdravnik napotovalec izda napotnico preko portala zVEM. Pacient prejme potrdilo o izdani eNapotnici, ki vsebuje njen identifikator. Pacient posreduje identifikator svoje veljavne IZDANE eNapotnice na eno izmed klinik, ki izvaja VZS opredeljen na eNapotnici. Napotnica je uvrščena v čakalno vrsto, pacientu pa je posredovan okvirni termin obravnave. Ko napotnica pride do konca čakalne vrste, je za njo kreirano naročilo s konkretnim terminom. Naročilo je nato realizirano z obravnavo pacienta. Na sliki 3.1 je prikazan oštevilčen diagram komunikacije sistemov pri omenjenem scenariju. Sledijo opisi interakcij sistemov posameznih komponent diagrama. Kjer v komunikaciji sodeluje tudi HOS sistem, je okvirno opisana tudi interna poslovna logika HOS sistema. Kronološko urejene interakcije med sistemi v opisanem scenariju so sledeče:

- **1** - Zdravnik napotovalec se prijavi v VS in tako pridobi svoj žeton varnostne sheme. Z žetonom se prijavi v portal zVEM, kjer ima omogočeno izdajo eNapotnic.
- **2** - Zdravnik napotovalec izda in podpiše napotnico na portalu zVEM. COS napotnico shrani v CRPP.
- **3** - Administrator HOS sistema se prijavi v VS in tako pridobi svoj žeton varnostne sheme.
- **4** - Pacient administratorju HOS sistema posreduje identifikator svoje eNapotnice. Na podlagi identifikatorja administrator v svoj HOS sistem pridobi podatke napotnice iz COS. COS podatke napotnice pridobi iz CRPP. Podatki napotnice so shranjeni v HOS sistem.
- **5** - Administrator uvrsti napotnico v čakalno vrsto. HOS v COS pošlje zahtevo za kreacijo naročila. Posredovan je okvirni termin naročila. Na podlagi kreiranega naročila v COS napotnica spremeni status v VPISANA, kar pomeni, da je uporaba napotnice drugim klinikam onemogočena.
- **6** - Napotnica pride do konca čakalne vrste. Administrator HOS sistema napotnico uvrsti na urnik. V COS je poslan zahtevek za prenaročanje na termin, ki ga je izbral administrator. V HOS sistemu je torej kreirano naročilo. Pacient je obveščen o terminu obiska.

- **7** - Pacient pride na obravnavo in je sprejet. V COS je poslan zahtevek za sprejem pacienta za napotnico, za katero je bilo v koraku 4 kreirano naročilo. V HOS je kreiran sprejem za naročilo.
- **8** - Obravnava je zaključena in kreiran je obisk. V COS se pošlje zahtevek za zaključek obravnave za napotnico, za katero je bilo v koraku 4 kreirano naročilo.



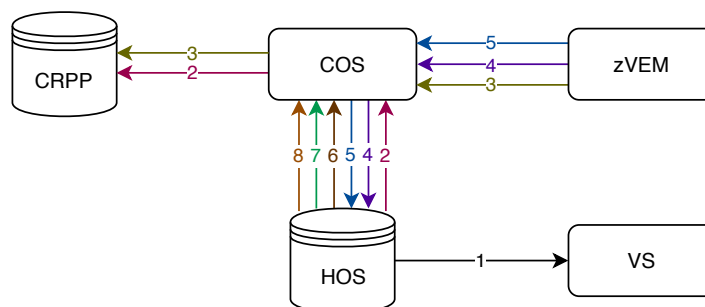
Slika 3.1: Enostavna shema gradnikov in interakcij sistemov eNaročanja med scenarijem, ki je opisan v razdelku 3.1.1. Interakcije med sistemi (ponazorjene s puščicami) so usmerjene od sistema, ki proži akcijo, proti sistemu, ki na akcijo odgovori. Številke na interakcijah, ki prikazujejo interakcije med sistemi, nakazujejo kronološko zaporedje dogodkov.

3.1.2 Izdaja napotnice iz HOS sistema in kreacija naročila preko portala zVEM

V tem razdelku opišemo scenarij, kjer je pacientu izdana eNapotnica v enem izmed HOS sistemov. Pacient prejme potrdilo o izdani eNapotnici, iz katerega pridobi identifikator eNapotnice, s pomočjo katerega se prijavi v portal zVEM. Na portalu pacient izbere eno izmed klinik, ki izvaja VZS opredeljen na eNapotnici (npr. tisto, ki ima najkrajšo čakalno dobo) in rezervira svoj (okvirni) termin. V HOS sistemu je napotnica uvrščena v čakalno vrsto. Ko napotnica pride do konca čakalne vrste, je za napotnico kreirano naročilo, ki je kasneje tudi realizirano. Na sliki 3.2 je prikazan oštevilčen diagram komunikacije sistemov pri omenjenem scenariju. Sledijo opisi interakcij sistemov posameznih komponent diagrama. Kjer v komunikaciji sodeluje tudi HOS sistem, je okvirno opisana tudi interna poslovna logika HOS sistema. Kronološko urejene interakcije med sistemi v opisanem

scenariju so sledeče:

- **1** - Zdravnik napotovalec se prijavi v VS in tako pridobi svoj žeton varnostne sheme. Z žetonom se prijavi v portal zVEM, kjer ima omogočeno izdajo eNapotnic.
- **2** - Zdravnik napotovalec izda in podpiše napotnico na portalu zVEM. COS napotnico shrani v CRPP.
- **3** - Pacient se prijavi v portal zVEM s pomočjo identifikatorja eNapotnice. Podatke eNapotnice COS pridobi iz CRPP.
- **4** - Pacient na portalu zVEM izbere eno izmed klinik, ki izvaja VZS določen na eNapotnici. Kreirano je prednaročilo.
- **5** - Pacient potrdi prednaročilo. Tako je uvrščen v ustrezno čakalno vrsto HOS sistema, ki je prevzel napotnico.
- **6** - Napotnica pride do konca čakalne vrste. Administrator HOS sistema napotnico uvrsti na urnik. V COS je poslan zahtevek za prenaročanje na termin, ki ga je izbral administrator. V HOS sistemu je kreirano naročilo. Pacient je obveščen o terminu obiska.
- **7** - Pacient pride na obravnavo in je sprejet. V COS je poslan zahtevek za sprejem pacienta za napotnico, za katero je bilo v koraku **5** kreirano naročilo. V HOS je kreiran sprejem za naročilo.
- **8** - Obravnava je zaključena in kreiran je obisk. V COS se pošlje zahtevek za zaključek obravnave za napotnico, za katero je bilo v koraku **5** kreirano naročilo.



Slika 3.2: Enostavna shema gradnikov in interakcij sistemov eNaročanja med scenarijem, ki je opisan v razdelku 3.1.2. Interakcije med sistemi (ponazorjene s puščicami) so usmerjene od sistema, ki proži akcijo, proti sistemu, ki na akcijo odgovori. Številke na interakcijah, ki prikazujejo interakcije med sistemi, nakazujejo kronološko zaporedje dogodkov.

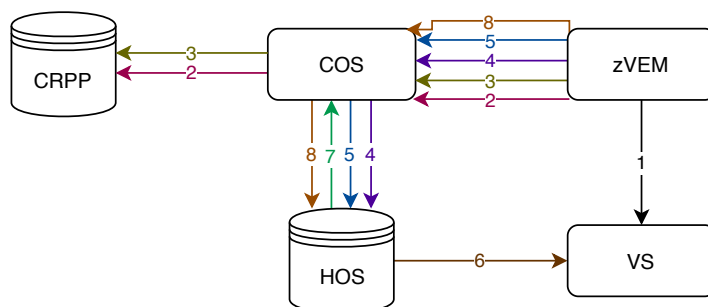
3.1.3 Izdaja napotnice, kreacija in preklic naročila preko portala zVEM

V tem razdelku opišemo scenarij, kjer pacientu zdravnik napotovalec izda napotnico preko portala zVEM. Pacient prejme potrdilo o izdani eNapotnici, ki vsebuje njen identifikator. S pomočjo identifikatorja in svoje ZZS številke se prijavi v portal zVEM. Na portalu izbere eno izmed ustanov, ki izvaja VZS opredeljen na eNapotnici. Kreira prednaročilo in ga potrdi. Tako je pacient uvrščen v čakalno vrsto ustanove. Ko njegova napotnica pride do konca čakalne vrste, jo administrator HOS sistema uvrsti na urnik in tako kreira naročilo. Pacient prejme obvestilo o terminu obiska. Pacient nato svoje naročilo prekliče preko portala zVEM. Na sliki 3.3 je prikazan oštevilčen diagram komunikacije sistemov pri omenjenem scenariju. Sledijo opisi interakcij sistemov posameznih komponent diagrama. Kjer v komunikaciji sodeluje tudi HOS sistem, je okvirno opisana tudi interna poslovna logika HOS sistema. Kronološko urejene interakcije med sistemi v opisanem scenariju so:

- **1** - Zdravnik napotovalec se prijavi v VS preko portala zVEM. Tako pridobi svoj žeton varnostne sheme.
- **2** - Zdravnik napotovalec izda in podpiše napotnico na portalu zVEM. COS napotnico shrani v CRPP.
- **3** - Pacient se prijavi v portal zVEM s pomočjo identifikatorja eNapotnice. Podatke

eNapotnice COS pridobi iz CRPP.

- 4 - Pacient na portalu zVEM izbere eno izmed klinik, ki izvaja VZS doloĦen na eNapotnici. Kreirano je prednaroĦilo.
- 5 - Pacient potrdi prednaroĦilo. Tako je uvrŦen v ustrezno Ħakalno vrsto HOS sistema, ki je prevzel napotnico.
- 6 - Administrator HOS sistema se prijavi v VS in tako pridobi svoj Ŧeton varnostne sheme.
- 7 - Napotnica pride do konca Ħakalne vrste. Administrator HOS sistema napotnico uvrsti na urnik. V COS je poslan zahtevek za prenaroĦanje na termin, ki ga je izbral administrator. V HOS sistemu je kreirano naroĦilo. Pacient je obveŦen o terminu obiska.
- 8 - Pacient prekliĦe naroĦilo preko portala zVEM. NaroĦilo je preklicano tudi v HOS sistemu.



Slika 3.3: Enostavna shema gradnikov in interakcij sistemov eĦarovanja med scenarijem, ki je opisan v razdelku 3.1.3. Interakcije med sistemi (ponazorjene s puŦicami) so usmerjene od sistema, ki proŦi akcijo, proti sistemu, ki na akcijo odgovori. Ŧtevilke na interakcijah, ki prikazujejo interakcije med sistemi, nakazujejo kronoloŦko zaporedje dogodkov.

3.2 Opis metod integracije med COS in HOS sistemi

Iz razdelka 3.1 je jasno razvidno dejstvo, da lahko metode integracije COS in HOS sistemov razdelimo glede na pobudnika komunikacije [7]. V razdelku 3.2.1 'HOS - COS'

bomo opisali metode, ki jih proži HOS, nanje pa ustrezno reagira in odgovori COS 'web service'. V razdelku 3.2.2 'COS - HOS' bomo opisali metode integracije, ki jih proži COS, nanje pa ustrezno reagira HOS 'web service'.

Pred samim opisom metod pa se moramo zavedati dejstva, da so bolnišnični (HOS) informacijski sistemi z vidika naročanja celovite informacijske rešitve. To pomeni, da imajo podprto naročanje tudi brez integracije v sistem eNaročanja. To je najboljše vidno v primeru izdaje in uporabe papirne napotnice za kreacijo naročila, kjer je napotnica prevzeta v fizični (papirnati) obliki. Za tak prevzem integracija s COS ni potrebna, kljub temu pa je scenarij uporabe napotnice za uporabnika HOS sistema identičen. Papirna napotnica je ročno prepisana v sistem, nato pa povezana na naročilo. Razlika je torej le v tem, da je potrebno podatke iz papirne napotnice ročno vpisati v sistem, medtem ko podatke elektronske napotnice v HOS sistem priskrbi preprost klic metode za pridobitev napotnice iz COS. Interna poslovna logika HOS informacijskega sistema mora biti iz tega vidika ekvivalentna za naročanje na podlagi papirne ali elektronske napotnice. Iz napisanega sledi, da je večina metod (izjema so metode za pridobivanje podatkov npr. pridobitev napotnice) opisanih v razdelku 3.2.1 uporabljenih izključno za sinhronizacijo stanja naročil in napotnic med HOS in COS sistemi. To pomeni, da je najprej izvedena poslovna logika HOS sistema, nato pa je novo stanje naročila in napotnice sinhronizirano v COS.

3.2.1 HOS - COS

Sledijo opisi metod integracije, kjer je pobudnik komunikacije HOS. Ob vsakem zahtevku je potrebno posredovati tudi žeton varnostne sheme. Opise metod bomo tudi smiselno razdelili glede na njihovo uporabo. Metode razdelimo na sledeče skupine:

■ Metodi za izdajo napotnice:

- `createReferral()`: izvede izdajo napotnice; ključni argument metode je napotnica v XML obliki. Ključni rezultat metode je identifikator napotnice;
- `signDocument()`: izvede podpis napotnice; ključni argument metode je z uporabniškim certifikatom podpisan XML napotnice;

■ Metodi za posodobitev in preklic napotnice:

- `updateReferral()`: izvede posodobitev napotnice; ključna argumenta metode

sta napotnica v XML obliki in identifikator napotnice. Po posodobitvi je napotnico potrebno tudi podpisati;

- `cancelReferral()`: izvede preklic napotnice; ključna argumenta metode sta identifikator napotnice in razlog preklica;

■ **Metode za kreacijo naročila in premik stanj napotnice:**

- `appointmentCreatedForReferral()`: izvede kreacijo naročila za napotnico; ključni argumenti metode so termin naročila, identifikator napotnice in identifikator naročila, ki ga določi HOS;
- `cancelAppointmentForReferral()`: izvede preklic naročila; ključni argument metode je identifikator naročila;
- `patientAdmittedForReferral()`: izvede sprejem pacienta; ključni argument metode je identifikator napotnice;
- `cancelAdmissionForReferral()`: izvede preklic sprejema pacienta; ključni argument metode je identifikator napotnice;
- `referralUsed()`: izvede realizacijo obiska; ključni argument metode je identifikator napotnice;
- `cancelReferralUsage()`: izvede preklic realizacije obiska; ključni argument metode je identifikator napotnice;
- `appointmentRescheduled()`: izvede prenaročanje naročila; ključni argument metode je identifikator naročila;

■ **Metode za manipulacijo priponk napotnice:**

- `createReferralAttachment()`: izvede kreacijo priponke; ključna argumenta metode sta priponka in identifikator napotnice;
- `updateReferralAttachment()`: izvede posodobitev priponke; ključna argumenta metode sta identifikator priponke in nova priponka;
- `deleteReferralAttachment()`: izvede izbris priponke; ključni argument metode je identifikator priponke;

■ **Metode za pridobitev podatkov iz centralnega sistema:**

- `getReferral()`: izvede pridobitev napotnice; ključni argument metode je identifikator napotnice;
- `getAppointment()`: izvede pridobitev naročila; ključni argument metode je identifikator naročila;
- `getAppointmentsForPatient()`: izvede pridobitev podatkov vseh aktualnih naročil za pacienta; ključni argument metode je pacientova ZZZS številka;
- `getReferralAttachment()`: izvede pridobitev priponke napotnice; ključna argumenta metode sta identifikator napotnice in identifikator priponke;
- `getReferralAttachments()`: izvede pridobitev vseh priponk napotnice; ključni argument metode je identifikator napotnice;
- `getGuidelineForProcedure()`: izvede pridobitev smernic za napotnico; ključni argument metode je identifikator napotnice;
- `getReferralList()`: izvede pridobitev administrativnih podatkov vseh veljavnih napotnic pacienta; ključni argument metode je pacientova ZZZS številka;

3.2.2 COS - HOS

Sledijo opisi metod integracije, kjer je pobudnik komunikacije COS. Metode razdelimo na sledeče skupine:

- **Metodi za pridobivanje podatkov o čakalnih vrstah:**
 - `getAppointmentsForProcedure()`: izvede pridobitev čakalne vrste in naročil; ključna argumenta metode sta identifikator ustanove in VZS;
 - `getRealisationsForProcedure()`: izvede pridobitev vseh realiziranih in nerealiziranih naročil današnjega dne; ključna argumenta sta identifikator ustanove in VZS;
- **Metode za naročanje preko portala zVEM:**
 - `getFreeSlotForProcedure()`: izvede pridobitev prostih (okvirnih) terminov; ključni argumenti metode so identifikator ustanove, VZS in število željenih terminov za vsako izmed stopenj nujnosti;
 - `createPrereservationForProcedure()`: izvede kreacijo predrezervacije naročila; ključni argumenti metode so identifikator napotnice, nujnost, VZS in administrativni podatki pacienta;

- `bookReservation`: izvede potrditev predrezervacije naročila; kreira naročilo oziroma vpis napotnice v čakalno vrsto; ključni argumenti metode so identifikator prednaročila in administrativni podatki napotnice;
- `cancelReservation()`: izvede preklic naročila; ključni argument metode je identifikator naročila;

3.3 Pomanjkljivosti sistema eNaročanje V1

Znano je, da so predolge čakalne vrste velik problem Slovenskega zdravstva. V ta namen smo razvili sistem, ki centralizira naročanje na državnem nivoju. Tipe zdravstvenih storitev smo večinoma dobro definirali z namenom boljšega pregleda nad čakalnimi dobami za določene zdravstvene storitve. Ideja izvira iz potrebe, da se pacientu na podlagi napotnice omogoči kar se da hiter prvi pregled pri zdravniku specialistu, pri tem pa seveda onemogoči goljufije, kot so preskakovanje čakalne vrste, prirejanje podatkov zdravstvenih ustanov o njihovih čakalnih vrstah itd.

Realizacija ideje je bila enostavna in hitro izvedena. Na podlagi napotnice kreiramo naročilo. Za napotnico lahko naenkrat obstaja samo eno naročilo. Na težave tega pristopa naletimo v trenutku, ko pogledamo dejanske prakse zdravstvenih ustanov. Ena izmed njih je ti. ambulantni padalec. Ambulantni padalec je pacient, ki pride v ambulantno po pomoč brez napotnice. Običajno so to nujne obravnave, zato pacient ni imel časa pridobiti napotnice pri osebemu zdravniku. Za pacienta je nemudoma kreirano naročilo, izvedena pa je tudi obravnava. Administracija se s pacientom dogovori, da bo napotnico poslal takoj, ko bo to mogoče. Ko administracija prejme napotnico, jo poveže na obisk. To pomeni, da se v tistem trenutku v COS kreira naročilo, pacient je sprejet in obravnava je zaključena. Podatki o času obiska, poslani v COS, so sicer praviloma točni, težava pa nastane v primeru, ko pacient pozabi naknadno poslati napotnico ali če administracija pozabi povezati napotnico na naročilo. V tem primeru naročilo obstaja v HOS sistemu, v COS pa ne. Z verzijo eNaročanjeV1 je bilo tako kar nekaj nezadovoljstva uporabnikov HOS informacijskih sistemov nad poslovno logiko eNaročanja.

3.3.1 Težave uporabnikov HOS sistemov

V pričujočem razdelku opišemo vsakodnevne težave HOS uporabnikov, ki so nastale zaradi implementacije eNaročanja v HOS sistemih. Le-te so sledeče:

- **Nekonsistentna stanja COS in HOS sistemov:** scenarijev, preko katerih HOS sistem lahko pride do nekonsistentnega stanja v primerjavi z COS sistemom (npr. napotnica je v HOS povezana na naročilo, v COS pa ima status IZDANA) je na žalost zelo veliko; lahko so posledica napake poslovne logike HOS ali COS sistema, lahko pa v nepravem trenutku pride do odpovedi delovanja prvega ali drugega sistema; rezultat nekonsistentnega stanja je onemogočeno delo za administracijo HOS sistema;
- **Razlog preklica eNapotnice:** zdravnik napotovalec prekliče IZDANO napotnico, pacient pa se kljub temu želi naročiti na obravnavo; v podatkih pridobljenih s klicem funkcije `getReferral()` ni bilo podatka o osebi in razlogu preklica napotnice;
- **Razlog preklica naročila:** pacient pozabi priti na obravnavo; naročilo je preklicano, napotnica pa spremeni status v IZDANA kljub temu, da razlog preklica ni bil upravičen oziroma naveden; napotnica je tako na voljo za ponovno uporabo;
- **Datum veljavnosti eNapotnice:** ob prvem sprejemu pacienta za večkratne napotnice se napotnici določi interval veljavnosti; ker podatka ni bilo specificiranega ob pridobivanju podatkov napotnice (`getReferral()`), je vsak HOS informacijski sistem sam računal interval veljavnosti; rezultat tega so nekonsistentna stanja;
- **Medicinski podatki:** mnogokrat je iz same napotnice težko razbrati dejanski razlog napotitve;
- **Umrli pacienti:** kljub temu, da ima CRPP podatke o pokojnih osebah, jim eNaročanje ni preklicalo naročila;
- **Pacienti iz tujine:** slovenski naslov je obvezen podatek pri izdaji napotnice (`createReferral()`);
- **Onemogočeno povezovanje napotnice pred zaključkom obravnave:** med obravnavo je pacienta nemogoče uvrstiti na ponovni pregled na podlagi večkratne napotnice, uporabljene na prvem pregledu; ko se obravnava zaključi, se napotnici spremeni status v IZDANA, šele takrat je omogočena kreacija novega naročila; zaradi te pomanjkljivosti uporabniki naknadno povezujejo napotnice na naročila; to pripelje do uporabniških napak;

3.3.2 Težave HOS sistemov pri implementaciji integracije

V pričujočem razdelku opišemo težave, na katere smo naleteli implementatorji eNaročanja v HOS sisteme. Le-te so sledeče:

- **Več ustanov z istim identifikatorjem:** obstajajo klinike, znotraj katerih je v uporabi več informacijskih sistemov; v bazi podatkov o izvajalcih (BPI) [8] je za vsako ustanovo določen samo en identifikator oz. številka; eNaročanje določi informacijski sistem, kjer je bilo kreirano naročilo na podlagi BPI številke, ki je vsebovana v identifikatorju naročila; na podlagi te pomanjkljivosti nekaterim klinikam eNaročanja ni možno implementirati;
- **Onemogočena komunikacija COS - HOS brez uporabnikovega zahtevka:** za izvedbo poizvedovalnih metod iz razdelka 3.1.5 potrebujemo žeton varnostne sheme; to pomeni, da se podatki iz COS v HOS lahko osvežijo samo na podlagi uporabniške zahteve; posledica je, da so podatki prikazani v HOS sistemih velikokrat neažurni; na podlagi tega so implementirane rešitve, kot je na primer osveževanje napotnice (`getReferral()`) ob vsaki akciji povezani z eNapotnico in eNaročilom; to predstavlja nepotrebno delo za oba sistema, kar privede do performančnih težav obeh sistemov;
- **Neprilagojenost poslovne logike eNaročanja na dosedanje scenarije naročanja:** eNaročanje dovoli kreacijo naročila samo na podlagi veljavne IZDANE eNapotnice; realni scenariji naročanja mnogokrat preferirajo kreiranje naročila pred pridobitvijo napotnice; rezultat tega je, da so v HOS sistemih na željo uporabnikov podprti tudi scenariji, ko se naroča brez napotnice, napotnica pa se na naročilo poveže kasneje; pojem povezovanje napotnice lahko tretiramo kot uskladitev stanja med COS in HOS sistemom; če povežemo napotnico na še nerealizirano naročilo, se mora napotnici status v COS spremeniti v VPISANA; če povezujemo napotnico na že realizirano naročilo oziroma zaključen obisk, se morajo v COS poslati podatki o kreiranju naročila, sprejemu in zaključku obiska; iz tega sledi, da status v HOS sistemih pripada naročilu, ne pa napotnici; razvezovanje napotnice in naročila tako rezultira v preklicu naročila v COS, v HOS pa ne; pridobivanje podatkov za pošiljanje realiziranih in nerealiziranih naročil (`getRealisationsForProcedure()`) je na tak način očitno nemogoče implementirati; na tem mestu velja omeniti tudi dejstvo, da je naročanje na podlagi papirnih napotnic še vedno podprto, saj mora biti

naročanje možno tudi ob izpadu centralnega sistema; torej je staro poslovno logiko naročanja nemogoče odstraniti iz HOS sistemov, saj mora delovati identično, kot je pred uveljavitvijo eNaročanja.

- **Pošiljanje čakalnih vrst:** na nekatere kontrolne preglede je naročeno zelo veliko število pacientov; količina podatkov, ki jih je potrebno poslati ob klicu metode `getAppointmentsForProcedure()` mnogokrat rezultira kot 'timeout' povezave; s tehničnega vidika HOS sistemov za to težavo ni rešitve;

4 Izboljšave sistema eNaročanja

V razdelku 3.3 smo opisali slabosti in pomanjkljivosti eNaročanja v prvotni obliki implementacije. Kot odgovor na pomanjkljivosti je bil 26. septembra 2017 sprejet zakon o spremembah in dopolnitvah Zakona o pacientovih pravicah (ZPacP-A) [9], na podlagi katerega so bile na centralnem sistemu naročanja (COS) implementirane spremembe, ki pa niso bile kompatibilne z obstoječimi implementacijami integracije na HOS sistemih. Torej govorimo o novi verziji eNaročanja, verziji 2, ki jo bomo opisali v sledečih razdelkih.

4.1 Izboljšave eNaročanja V2

V tem razdelku opišemo vsebinske spremembe eNaročanja [10], ki neposredno vplivajo na poslovno logiko HOS sistemov ob prehodu na novo verzijo. Le te so sledeče:

- **Razlog preklica eNapotnice.** Ob stornu eNapotnice je potrebno navesti razlog preklica napotnice. Ob klicu funkcije `getReferral()` je za preklicano napotnico evidentiran tudi razlog preklica in oseba, ki je napotnico preklicala.
- **Razlog preklica naročila.** Ob preklicu naročila je potrebno navesti tudi razlog preklica. Če razlog ni opravičljiv (npr. pacient pozabi priti na obravnavo) na-

potnica spremeni status v NEIZKORIŠĀENA, kar pomeni, da napotnica ni veĀ primerna za nadaljno uporabo.

- **Datum veljavnosti eNapotnice.** Ob prvem sprejemu za napotnico (klic funkcije `patientAdmittedForReferral()`) se VEĀKRNATNI napotnici doloĀi interval veljavnosti, ki je izraĀunan na podlagi datuma sprejema in števila mesecev veljavnosti napotnice. Interval veljavnosti je mogoĀe pridobiti z uporabo funkcije `getReferral()`.
- **VeĀ ustanov z isto BPI številko.** Poleg BPI identifikatorja ustanove je za potrebe eNaroĀanja ustanov, kjer sta v uporabi dva informacijska sistema, uveden identifikator, ki je unikatni za kliniko in informacijski sistem. Sestavljen je iz BPI številke ustanove in Ārke A ali B. Primer: Think!Clinical je eden izmed dveh informacijskih sistemov na Kliniki za nuklearno medicino. BPI identifikator Klinike za nuklearno medicino je 50016, Think!Clinical pa kot drugi informacijski sistem pridobi svoj unikatni identifikator 50016B. Identifikator (poimenovan 'healthCare-ProviderSpecificIndex') je mogoĀe videti v uporabi v naslednjem razdelku, kjer je na izpisu 4.6 prikazana XML struktura sporoĀila za preklic napotnice.
eNaroĀanje na podlagi identifikatorja naroĀila prepozna informacijski sistem in ustanovo, kjer je bilo naroĀilo kreirano. Zato potrebujemo tudi logiko, preko katere COS prepozna informacijski sistem v primeru, ko za isti BPI identifikator obstajata dva informacijska sistema. Identifikator naroĀila je 13-mestno število, ki je sestavljeno iz 5-mestne BPI številke ustanove, sledita zadnji 2 števki trenutnega leta (v Āasu pisanja diplome torej 19) na koncu pa je 6-mestno število, ki se periodiĀno poveĀuje s kreacijo vsakega naslednjega naroĀila in je ponastavljeno na zaĀetno vrednost 0 ob zaĀetku leta. Za potrebe eNaroĀanja veĀih informacijskih sistemov, je v HOS informacijskih sistemih, ki imajo svoj identifikator, ki se konĀa z 'B' uvedeno pravilo: namesto zadnjih dveh števk trenutnega leta, uporabimo število, ki ga pridobimo, Āe številu, ki je sestavljeno iz zadnjih dveh števk trenutnega leta prištejemo število 50. Primer: HOS informacijski sistem 50016A kreira prvo naroĀilo v letu 2019 z identifikatorjem 500161900000000, med tem ko informacijski sistem 50016B prvo naroĀilo kreira z identifikatorjem 500166900000000.
- **Pristop k storitvi preverjanja umrlih pacientov.** COS na podlagi podatkov iz CRPP periodiĀno preklicuje naroĀila umrlim pacientom (klic funkcije `cancelReservation()` spletne storitve HOS). Napotnice spremenijo status v NEIZKORIŠĀENA

(če za napotnico še ne obstaja obisk) oziroma IZKORIŠČENA.

- **Podpora za šifro občine 000-Tujina.** Omogočena izdaja napotnic pacientom brez slovenskega naslova.

4.2 Vzorčna primera izvedbe nadgradnje na novo verzijo

V pričujočem razdelku bomo opisali dva vzorčna primera nadgradnje HOS informacijskega sistema ob prehodu na novo verzijo eNaročanja. Opisali bomo nadgradnji sledečih funkcij:

- `getAppointmentForProcedure()` spletne storitve HOS, ki smo jo predhodno opisali v razdelku 3.2.2;
- `cancelReferral()` spletne storitve COS, ki smo jo predhodno opisali v razdelku 3.2.1;

Pred opisom je potrebno vsaj na kratko predstaviti tehnologije, katerih uporabo bomo prikazali v nadaljevanju.

Spletne storitve in njihovi odjemalci za komunikacijo uporabljajo protokol SOAP (angl. *Simple Object Access Protocol*) [11]. SOAP je aplikacijski komunikacijski protokol, ki tipično deluje nad aplikacijskim komunikacijskim protokolom HTTP. SOAP vsebina HTTP sporočila je niz v XML obliki, ki je sestavljen iz ovojnice (angl. *envelope*), ki opcijsko vsebuje glavo (angl. *header*) in telo (angl. *body*). Slednji nosi vsebino sporočila in opcijsko atribut napake (angl. *fault*), ki je definiran takrat, ko pri procesiranju zahtevka pride do napake. Primer enostavnega SOAP sporočila se nahaja na izpisu 4.1.

```
1 <?xml version="1.0"?>
2 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
3   <soap:Header>
4     <!--...-->
5   </soap:Header>
6   <soap:Body>
7     <!--payload-->
8     <soap:Fault>
9       <!--error-->
10    </soap:Fault>
11  </soap:Body>
12 </soap:Envelope>
```

Izpis 4.1: Enostavna struktura primera SOAP sporočila.

Spletne storitve SOAP so opisane z (angl. *Web Services Description Language*) datotekami [12]. WSDL datoteka specificira lokacijo, metodo in obliko SOAP sporoĀil. V naŖem primeru so oblike vsebine SOAP sporoĀil definirane v loĀenih XSD (angl. *XML Schema Definition*) datotekah zaradi boljŖe preglednosti.

Aplikacijski streŖnik Think!Clinical je implementiran v programskem jeziku Java. Za kompilacijo projekta (angl. *build*) je uporabljeno orodje Maven. Slednje avtomatizira kompilacijo veĀ-modularnih projektov [13]. Na voljo ima tudi vmesnike oziroma orodja (angl. *plugin*). Uporabo enega izmed njih bomo prikazali v sledeĀih razdelkih.

Komunikacija med COS in HOS sistemi torej poteka preko SOAP protokola, ki uporablja XML strukturo svojih sporoĀil. Z novo verzijo eĀaroĀanja so se za potrebe novitet spremenile definicije funkcij in oblike podatkov, ki se poŖiljajo. PoslediĀno se tako spremenijo WSDL in XSD datoteke. Spremembe definira implementator centralnega sistema (COS), ker je HOS sistemov mnogo, COS pa je samo en. Ko nadgrajujemo integracijo z neko spletno storitvijo, je tako najprej potrebno pridobiti WSDL datoteke, ki opisujejo spletno storitev in XSD datoteke, ki predstavljajo XML strukturo sporoĀil.

4.2.1 Nadgradnja funkcije `getAppointmentForProcedure()`

V priĀujoĀem razdelku bomo opisali nadgradnjo funkcije `getAppointmentsForProcedure()` (pridobivanje Āakalnih vrst) HOS spletne storitve. Pri tej nadgradnji uporabnik aplikacije ne zazna sprememb, nadgradnja pa se nahaja izkljuĀno na aplikacijskem streŖniku. Funkcija se je z eĀaroĀanjem V2 spremenila tako, da je bila dodana funkcionalnost paginacije (poŖiljanje rezultata v veĀih korakih).

Funkcija je bila definirana tudi v eĀaroĀanju V1, torej je WSDL datoteka na tem mestu nespremenjena. Na izpisu 4.2 je prikazan izsek WSDL datoteke, ki definira opisano metodo. Spremenjena je oblika XML sporoĀil. Na izpisu 4.3 je prikazana XSD struktura 'request' in 'response' sporoĀila. Na 'request' sporoĀilu je dodan podatek o indeksu strani, ki se mora poslati, na 'response' sporoĀilu pa je dodana struktura 'PagingInfo', ki vsebuje podatke potrebne za implementacijo paginacije na strani COS. Z uporabo Maven 'jaxb2-maven-plugin' [14] iz XSD datotek generiramo javanski razred, ki ponazarja objekt, ki ustreza XSD definiciji. Na izpisu 4.4 je prikazan primer novo generiranega podrazreda PagingInfo. Na izpisu 4.5 je prikazana uporaba objekta paginacije.

1 `<s:element name="GetAppointmentsRequest">`

```

2   <s:complexType>
3   <s:sequence>
4     <s:element minOccurs="1" name="version" type="s:int" />
5     <s:element minOccurs="0" name="request" type="s:string" />
6   </s:sequence>
7 </s:complexType>
8 </s:element>
9 <s:element name=" GetAppointmentsResponse">
10  <s:complexType>
11  <s:sequence>
12    <s:element minOccurs="0" name="response" type="s:string" />
13  </s:sequence>
14 </s:complexType>
15 </s:element>
16 </s:schema>
17 </wsdl:types>
18 <wsdl:message name=" GetAppointmentsSoapIn">
19   <wsdl:part name="parameters" element="tns: GetAppointmentsRequest" />
20 </wsdl:message>
21 <wsdl:message name=" GetAppointmentsSoapOut">
22   <wsdl:part name="parameters" element="tns: GetAppointmentsResponse" />
23 </wsdl:message>
24 <wsdl:portType name="HOSSoap">
25   <wsdl:operation name=" GetAppointments">
26     <wsdl:input message="tns: GetAppointmentsSoapIn" />
27     <wsdl:output message="tns: GetAppointmentsSoapOut" />
28   </wsdl:operation>
29 </wsdl:portType>

```

Izpis 4.2: Izsek WSDL datoteke, ki definira metodo `getAppointmentsForProcedure()` spletne storitve HOS. Na vrhu izpisa je definirana vsebina vhodnega sporočila 'GetAppointmentsForProcedureRequest', ki ima definirana dva elementa: 'int version' in 'String request'. 'Version' je uporabljen za potrebe prehoda na novo verzijo in ga bomo obdelali v razdelku 4.3, 'request' pa je tipa 'String', ki mora zadoščati XML shemi ponazorjeni na izpisu 4.3.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
3   <xs:element name="GetAppointmentsRequest">
4     <xs:complexType>
5       <xs:sequence>

```

```

6 <xs:element name="MedicalFacilityCode" type="xs:string" />
7 <xs:element name="MedicalFacilitySpecificCode" type="xs:string" />
8 <xs:element name="MedicalProcedureCode" type="xs:string" />
9 <xs:element name="FromDate" type="xs:dateTime" />
10 <xs:element name="PageNumber" type="xs:integer" nillable="true" />
11 </xs:sequence>
12 </xs:complexType>
13 </xs:element>
14
15 <xs:element name="GetAppointmentsResponse">
16 <xs:complexType>
17 <xs:sequence>
18 <xs:element name="MedicalFacilitySpecificCode" type="xs:string" />
19 <xs:element name="PagingInfo" nillable="true">
20 <xs:complexType>
21 <xs:sequence>
22 <xs:element name="PageNumber" type="xs:integer" />
23 <xs:element name="PageSize" type="xs:integer" />
24 <xs:element name="RemainingNumber" type="xs:integer" />
25 <xs:element name="TotalNumber" type="xs:integer" />
26 </xs:sequence>
27 </xs:complexType>
28 </xs:element>
29 <xs:element name="PatientAppointments">
30 <!-- ... -->
31 </xs:element>
32 </xs:sequence>
33 </xs:complexType>
34 </xs:element>
35 </xs:schema>

```

Izpis 4.3: Izsek XSD datoteke, ki ponazarja XML strukturo vhodnega ('request') in izhodnega ('response') sporočila pri uporabi funkcije `getAppointmentsForProcedure()`. Na 'request' sporočilo je v primerjavi z eNaročanjem V1 dodan podatek o zahtevani strani rezultata ('pageNumber'), na 'response' sporočilo pa je dodana struktura 'PagingInfo'. Sheme preostalih podatkov (naročila itd.) so zaradi preglednosti izpuščene.

```

1 @XmlElement(name = "GetAppointmentsResp")
2 public class GetAppointmentsResponse
3 {
4     @XmlAccessorType(XmlAccessType.FIELD)

```

```

5  @XmlType(name = "", propOrder = {
6      "pageNumber",
7      "pageSize",
8      "remainingNumber",
9      "totalNumber"
10 })
11 public static class PagingInfo {
12     @XmlElement(name = "PageNumber", required = true)
13     private BigInteger pageNumber;
14     @XmlElement(name = "PageSize", required = true)
15     private BigInteger pageSize;
16     @XmlElement(name = "RemainingNumber", required = true)
17     private BigInteger remainingNumber;
18     @XmlElement(name = "TotalNumber", required = true)
19     private BigInteger totalNumber;
20
21     public void setPageNumber(final BigInteger value) {
22         pageNumber = value;
23     }
24     public void setPageSize(final BigInteger value) {
25         pageSize = value;
26     }
27     public void setRemainingNumber(final BigInteger value) {
28         remainingNumber = value;
29     }
30     public void setTotalNumber(final BigInteger value) {
31         totalNumber = value;
32     }
33 }
34 //...
35 }

```

Izpis 4.4: Izsek razreda `GetAppointmentsResponse.java`, ki ga je generiral 'jaxb2-maven-plugin'. `GetAppointmentsResponse.java` vsebuje nov podrazred `PagingInfo`.

```

1 long appointmentsCount = getAppointmentsForProcedureCount(
2     input.getMedicalFacilityCode(),
3     input.getMedicalProcedureCode(),
4     fromDate);
5
6 if (appointmentsCount > APPOINTMENTS_FOR_PROCEDURE_BATCH_SIZE){
7     GetAppointmentsResponse.PagingInfo pagingInfo = new GetAppointmentsResponse.PagingInfo();

```

```

8   pagingInfo.setPageNumber(pageNumber);
9   pagingInfo.setPageSize(appointments.getPatientAppointment().size());
10  pagingInfo.setRemainingNumber(
11      appointmentsCount - appointments.getPatientAppointment().size() -
12      APPOINTMENTS_BATCH_SIZE * (pageNumber - 1));
13  pagingInfo.setTotalNumber(BigInteger.valueOf(appointmentsCount));
14  response.setPagingInfo(pagingInfo);
15  }
16  return response;

```

Izpis 4.5: Implementacija funkcionalnosti paginacije. V primeru, ko število vseh naročil, ki jih je potrebno poslati, presega maksimalno velikost enega sporočila, se smiselno napolni nov objekt `PagingInfo`.

4.2.2 Nadgradnja funkcije `cancelReferral()`

V pričujočem razdelku bomo opisali nadgradnjo funkcije `cancelReferral()` in prikazali način uporabe spremenjene funkcionalnosti. Ob preklicu napotnice mora uporabnik, ki preklicuje napotnico, vnesti razlog preklica napotnice in opcijsko tudi opis razloga preklica. WSDL datoteka na tem mestu ostane nespremenjena, ker je bila funkcija identično definirana že v eNaročanju V1. V WSDL datoteki je podobno kot v primeru iz prejšnjega razdelka v `'CancelReferralRequest'` sporočilu definiran `'String request'` in `'int version'`. `'Request'` mora biti v XML obliki in zadoščati shemi definirani v pripadajoči XSD datoteki, prikazani na izpisu 4.6. V XSD datoteko so glede na staro verzijo eNaročanja dodana polja za edinstveni identifikator informacijskega sistema v ustanovi (`'healthCareProviderSpecificIndex'`), šifra razloga preklica in opis razloga preklica. XSD datoteko pretvorimo v javanski objekt z pomočjo `'jaxb2-maven-plugin'` in tako pridobimo objekt na izpisu 4.7. Na klientu implementiramo enostavno formo za vnos razloga preklica napotnice. Forma je prikazana na sliki 4.1. Iz forme pridobimo podatke, ki jih uporabimo za polnjenje objekta iz ispisa 4.7. Implementacija polnjenja objekta z novimi parametri je prikazana na izpisu 4.8.

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" >
3  <xs:element name="CancelReferralRequest" >
4  <xs:complexType>
5  <xs:sequence>
6  <xs:element name="HealthCareProviderIndex" type="xs:string" />

```

```

7 <xs:element name="ReferralUniqueIdentifier" type="xs:string" />
8 <xs:element name="CancelledBy" type="xs:string" />
9
10 <xs:element name="HealthCareProviderSpecificIndex" type="xs:string" />
11 <xs:element name="CancellationReasonCode" type="xs:string" />
12 <xs:element name="CancellationReasonDescription" type="xs:string" />
13 </xs:sequence>
14 </xs:complexType>
15 </xs:element>
16 </xs:schema>

```

Izpis 4.6: XSD shema 'request' sporočila pri uporabi metode `cancelReferral`. Glede na prejšnjo verzijo so dodana polja za unikaten identifikator za kliniko in informacijski sistem (`healthCareProviderSpecificIndex`), šifra razloga preklica in opis razloga preklica.

```

1 @XmlElement(name = "CancelReferralRequest")
2 public class CancelReferralRequest {
3     @XmlElement(name = "HealthCareProviderIndex", required = true)
4     private String healthCareProviderIndex;
5     @XmlElement(name = "ReferralUniqueIdentifier", required = true)
6     private String referralUniqueIdentifier;
7     @XmlElement(name = "CancelledBy", required = true)
8     private String cancelledBy;
9     @XmlElement(name = "CancellationReasonCode", required = true)
10    private String cancellationReasonCode;
11    @XmlElement(name = "CancellationReasonDescription")
12    private String cancellationReasonDescription;
13    @XmlElement(name = "HealthCareProviderSpecificIndex")
14    private String healthCareProviderSpecificIndex;
15
16    public void setHealthCareProviderIndex(final String value) {
17        healthCareProviderIndex = value;
18    }
19    public void setReferralUniqueIdentifier(final String value) {
20        referralUniqueIdentifier = value;
21    }
22    public void setCancelledBy(final String value) {
23        cancelledBy = value;
24    }
25    public void setCancellationReasonCode(final String value) {
26        cancellationReasonCode = value;

```

```

27     }
28     public void setCancellationReasonDescription(final String value) {
29         cancellationReasonDescription = value;
30     }
31     public void setHealthCareProviderSpecificIndex(final String value) {
32         healthCareProviderSpecificIndex = value;
33     }
34 }

```

Izpis 4.7: Razred `CancelReferralRequest.java`, ki ga je generiral 'jAXB2-maven-plugin'. Razred `CancelReferralRequest.java` vsebuje nove podatke za unikatni identifikator za kliniko in informacijski sistem ('`healthCareProviderSpecificIndex`'), šifro razloga preklica in opis razloga preklica.

Slika 4.1: Forma namenjena vnosu razloga in opisu preklica napotnice.

```

1 final CancelReferralRequest request = new CancelReferralRequest();
2
3 request.setHealthCareProviderIndex(credDto.getOrgBPI());
4 request.setCancelledBy(getUserBpi(credDto));
5 request.setReferralUniqueIdentifier(requestDto.getReferralUniqueIdentifier());
6
7 request.setCancellationReasonCode(requestDto.getCancellationReasonCode());
8 request.setCancellationReasonDescription(requestDto.getCancellationDescription());

```

```
9 request.setHealthCareProviderSpecificIndex(  
10     specificIndexProvider.provide(credDto.getOrgBPI()));  
11 return request;
```

Izpis 4.8: Implementacija polnjenja novih atributov razreda `CancelReferralRequest.java`.

4.3 Način prehoda

V tem razdelku opišemo način prehoda med eNaročanjem V1 in eNaročanjem V2. V razdelku 4.2 smo prikazali izsek WSDL datoteke, ki definira metodo `getAppointmentsForProcedure()` na izpisu 4.2. Opazimo, da ima razred `GetAppointmentsForProcedureRequest` dva parametra, in sicer:

- **int version:** število, ki ponazarja verzijo eNaročanja po kateri poteka trenutna komunikacija,
- **String request:** niz, ki mora biti oblike določene v XSD datotekah.

Slednje opažanje lahko apliciramo tudi na vse ostale metode integracije, tako za komunikacijo 'COS-HOS', kot za komunikacijo 'HOS-COS'. Centralni sistem (COS) je v prehodnem obdobju sprejemal (komunikacija 'HOS-COS') 'requeste' za obe obstoječi verziji (1 in 2). Oblika XML sporočil je seveda morala sovpadati z XSD shemami, ki so pripadale uporabljeni verziji eNaročanja. Torej je ob prehodu na novo verzijo HOS sistem samo spremenil vrednost 'version' atributa v SOAP 'request' sporočilu in pošiljal 'request' sporočila v novi XML obliki. Komunikacija 'COS-HOS' pa ne podpira dveh verzij komunikacije hkrati. Zato je bil za potrebe prehoda na novo verzijo implementiran poseben tip sporočila, definiran v svoji XSD datoteki. To sporočilo je namenjeno izključno za posredovanje informacije, da je sistem pripravljen na prehod na novo verzijo. XSD shema sporočila 'Error' je prikazana na izpisu 4.9. Iz XSD datoteke generiramo javanski objekt, ki je prikazan na izpisu 4.10. Ob prehodu na novo verzijo eNaročanja je HOS strežnik ob prvem 'requestu', ki je imel 'version' atribut enak 1, produciral napako v zgoraj omenjeni obliki. To pomeni, da je namesto standardnega odgovora (SOAP sporočilo z ustreznim 'body' elementom, ki vsebuje pričakovan XML) vrnil SOAP sporočilo, ki v 'body' elementu nima pričakovanih podatkov, temveč element 'Fault', ki vsebuje XML predhodno definirane napake. Prikaz uporabe objekta za posredovanje informacije o novi verziji je prikazan na izpisu 4.11. Ob posredovani napaki je COS spremenil svoje delovanje in začel pošiljati sporočila z verzijo 2.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
3   <xs:element name="Error">
4     <xs:complexType>
5       <xs:sequence>
6         <xs:element name="ErrorMessage" type="xs:string"/>
7         <xs:element name="ErrorNumber" type="xs:string"/>
8       </xs:sequence>
9     </xs:complexType>
10  </xs:element>
11 </xs:schema>

```

Izpis 4.9: XSD shema 'Error' sporočila, ki se uporablja za obveščanje centralnega sistema o prehodu na novo verzijo HOS sistema.

```

1 @XmlAccessorType(XmlAccessType.FIELD)
2 @XmlType(name = "", propOrder = {
3     "errorMessage",
4     "errorNumber"
5 })
6 @XmlRootElement(name = "Error")
7 public class Error {
8     @XmlElement(name = "ErrorMessage", required = true)
9     private String errorMessage;
10    @XmlElement(name = "ErrorNumber", required = true)
11    private String errorNumber;
12
13    public void setErrorMessage(final String value) {
14        errorMessage = value;
15    }
16    public void setErrorNumber(final String value) {
17        errorNumber = value;
18    }
19 }

```

Izpis 4.10: Razred Error.java, ki ga je generiral 'jaxb2-maven-plugin'.

```

1 final Error error = new Error();
2 error.setErrorMessage(hosIncompatibleVersionErrorMessage);
3 error.setErrorNumber(hosIncompatibleVersionErrorNumber);
4
5 try{

```

```
6     final SoapFaultDetail soapFaultDetail = fault.addFaultDetail();
7     final JAXBContext context = JAXBContext.newInstance(Error.class);
8     final Marshaller marshaller = context.createMarshaller();
9     marshaller.marshal(error, soapFaultDetail.getResult());
10  }
11  catch (final JAXBException e){
12      throw ThrowableUtil.asRuntime(e);
13  }
```

Izpis 4.11: Kreacija objekta `Error.java` in nastavitev 'SOAP Fault' segmenta z vsebino kreiranega omenjenega objekta.

5 Zaključek

eNaročanje je danes razmeroma dobro uveljavljen sistem elektronskega naročanja na zdravstvene storitve. Sistem povezuje informacijske sisteme zdravstvenih ustanov izvajalcev zdravstvenih dejavnosti (HOS) z centralnim sistemom naročanja (COS), ki za hrambo svojih podatkov uporablja centralni register podatkov o pacientih (CRPP) in tako centralizira podatke naročanja na državni ravni. Za avtentikacijo in avtorizacijo skrbi varnostna shema (VS). Z možnostjo naročanja preko spleta, ki ga eNaročanje omogoča preko portala zVEM, je prebivalstvu omogočena izbira željene napotne ustanove, ki jo lahko določijo tudi na podlagi posredovane pričakovane čakalne dobe. Uvedena so nova pravila naročanja, ki onemogočajo goljufije (preskakovanje čakalne vrste itd.) in hkrati optimizirajo delovne procese izvajalcev zdravstvenih dejavnosti.

eNaročanje je v svoji prvotni obliki izvedbe (verzija 1) v informacijske sisteme izvajalcev zdravstvenih dejavnosti in posledično njihovim uporabnikom prineslo kar nekaj težav in posledično tudi otežilo delovni proces pri realizaciji posameznih naročil. Omenjene težave smo opisali v poglavju 3, nato pa smo v poglavju 4 opisali prehod in izboljšave nove verzije eNaročanja, ki je bila uvedena na podlagi zakona o spremembah in

dopolnitvah zakona o pacientovih pravicah (ZPacP-A).

Nova verzija eNaročanja na žalost ni vsebovala rešitve za težavi, ki obstojata od uveljavitve eNaročanja pa vse do današnjega dne. Težavi sta naslednji:

- onemogočena strežniška komunikacija 'HOS-COS' brez uporabniškega žetona varnostne sheme;
- onemogočen sočasen obstoj večih aktualnih naročil oziroma onemogočen obstoj naročila brez napotnice;

Posledica prve težave je onemogočena asinhrona komunikacija med sistemi, ki bi v določenih primerih bistveno pohitrila delovanje HOS sistema. Prikazovanje podatkov o naročilih in napotnicah bi bilo na tak način mogoče implementirati tako, da uporabniku nebi bilo potrebno eksplicitno zahtevati določene vsebine iz centralnega sistema, ampak bi za prikazovanje ažurnih podatkov skrbel aplikacijski strežnik HOS informacijskega sistema. Posledica drugega dejstva pa je precej bolj kompleksna, njena rešitev pa je nujno potrebna za uspešen bodoči razvoj projekta eNaročanje. Status naročila je v eNaročanju vsebovan na podatkovni strukturi napotnice, kljub temu, da je naročilo tisto, ki v svojem ciklu spreminja status. Med sistemoma se fundamentalno razlikuje tudi relacija oz. uporaba napotnice. COS kreira naročilo na podlagi napotnice. HOS kreira naročilo, na katero je lahko povezana napotnica, ni pa to pogoj za obstoj naročila. Napotnico je na naročilo možno v katerem koli trenutku povezati (seveda samo napotnice, ki so v času predvidenega obiska veljavne) ali razvezati. Posledica je torej ta, da HOS informacijski sistemi vodijo statuse na podatkovni strukturi naročila, zato je poslovna logika, ki skrbi za usklajeno stanje med sistemoma 'ahilova tetiva' implementacije integracije med sistemoma na strani HOS. Uporabnikom to povzroča nemalo težav in zmedo pri opravljanju njihovega dela. Za pacienta, ki ima veljavno večkratno napotnico, je mnogokrat kreiranih več naročil hkrati. Napotnica pa je lahko povezana samo na eno aktualno naročilo naenkrat. Torej morajo administratorji sistema naknadno povezovati napotnice na naročila, kar pa ni optimalno. Iz omenjenega dejstva sledi, da eNaročanje sploh ne pozna naročil HOS sistemov, ki nimajo povezane napotnice. Omogočiti je torej potrebno kreacijo naročil brez napotnice in obstoj večih aktualnih naročil za isto VEČKRATNO napotnico hkrati. To je cilj, h kateremu stremimo tako razvijalci kot tudi uporabniki sistema eNaročanja.

LITERATURA

- [1] Navodilo za uveljavljanje pravic z napotnico, Dosegljivo: <http://www.zzzs.si/Zzzs/info/egradiva.nsf/0/38ff11568d0e77c7c12577fb00426e00>, [Dostopano: 5. 2. 2019].
- [2] Centralni register podatkov o pacientu, Dosegljivo: <http://www.ezdrav.si/category/projekti/crpp/>, [Dostopano: 5. 2. 2019].
- [3] Think!EHR platform, Dosegljivo: <http://www.marand.com/thinkehr>, [Dostopano: 5. 2. 2019].
- [4] What is openEHR, Dosegljivo: https://www.openehr.org/what_is_openehr, [Dostopano: 5. 2. 2019].
- [5] Varnostna shema eZdravje, Dosegljivo: <https://storitve.ezdrav.si/VS.web/>, [Dostopano: 5. 2. 2019].
- [6] Portal zVEM, Dosegljivo: <https://zvem.ezdrav.si/domov>, [Dostopano: 15. 1. 2019].
- [7] Paraman Filip, Web service specification V2.12, gradivo ni javno dostopno, lastnik gradiva je NIJZ (2013).
- [8] Baza podatkov izvajalcev zdravstvenih dejavnosti, Dosegljivo: https://partner.zzzs.si/wps/portal/portali/aizv/e-poslovanje/baza_podatkov_o_izvajalcih_zdravstvene_dejavnosti!/ut/p/z1/04_Sj9CPykssy0xPLMnMz0vMAfIjo8ziTQxdPd2N_Q08DUz8jQwcjVxNPI18AgwNTMz1C7IdFQF0Zx4Z/, [Dostopano: 20. 2. 2019].
- [9] Zakon o spremembah in dopolnitvah Zakona o pacientovih pravicah – ZPacP-A , Dosegljivo: <http://pisrs.si/Pis.web/pregledPredpisa?id=ZAK04281#>, [Dostopano: 5. 2. 2019].

- [10] Siniša Koščina, Funkcionalna specifikacija rešitve eNaročanje v2.09, gradivo ni javno dostopno, lastnik gradiva je NIJZ (2013).
- [11] XML SOAP, Dosegljivo: https://www.w3schools.com/xml/xml_soap.asp, [Dostopano: 1. 3. 2019].
- [12] XML WSDL, Dosegljivo: https://www.w3schools.com/xml/xml_wsd1.asp, [Dostopano: 1. 3. 2019].
- [13] Apache Maven, Dosegljivo: <https://maven.apache.org/>, [Dostopano: 1. 3. 2019].
- [14] Maven jaxb2-maven-plugin, Dosegljivo: <https://mvnrepository.com/artifact/org.codehaus.mojo/jaxb2-maven-plugin>, [Dostopano: 1. 3. 2019].