

Pregledni znanstveni članek ■

## Dvonivojski informacijski sistem po modelu openEHR za podporo kliničnemu zdravljenju

## Two-level openEHR-based Information System for Supporting Clinical Treatment

---

Instituciji avtorjev: Vortal d.o.o., Domžale, Slovenija (BM); Marand d.o.o., Ljubljana, Slovenija (TG).

Kontaktna oseba: Branko Mihovilovič, Vortal d.o.o., Mala Loka 27, 1230 Domžale. e-pošta: branko.mihovilovic@siol.com

Prejeto: 02.11.2011. Sprejeto: 02.12.2011.

**Branko Mihovilovič, Tomaž Gornik**

**Izvleček.** Večina današnjih informacijskih sistemov (IS) za podporo procesom kliničnega zdravljenja pacientov je zasnovana na enonivojski metodologiji, kjer so informacijski koncepti in koncepti znanja vgrajeni v enovite objektne in podatkovne modele. Veliko število domenskih konceptov v medicini, predvsem pa njihova velika zapletenost in nestanovitnost, povzročata velike stroške vzdrževanja enonivojskega sistema, zamenjave delov ali zamenjave celotne aplikacije. V dvonivojski metodologiji je IS zasnovan na informacijskem modelu (prvi nivo), v realnem času pa ga poganjajo koncepti domenskega znanja (medicina, mikrobiologija, itd.), ki jih predstavljajo arhetipi (drugi nivo). Za prvi nivo skrbijo razvijalci informacijskih sistemov, na drugem nivoju pa lahko domenski uporabniki (zdravniki, medicinske sestre idr.) v polni meri izkoristijo svoj potencial pri implementaciji domenskega znanja.

**Abstract.** Most information systems (IS) in medicine are built using single-level methodologies, where both information and knowledge concepts are built into one level of object and data models. Given the large number of concepts in medicine, their complexity and a high rate of definitional change, systems based on such model are expensive to maintain. Two-level IS's are built from the information models (first level) only, at runtime, those systems are driven by the knowledge concepts (medicine, microbiology etc.), defined in the archetypes (second level). This way, information technology specialists can develop the system more quickly, while the archetypes are authored directly by domain specialists (physicians of various specialties, nurses etc.), who can thus make full use of their knowledge potential in the IS implementation.

■ **Infor Med Slov:** 2011; 16(2): 15-29

## Uvod

Pacientova zdravstvena kartoteka ni samo kopičenje z zdravjem povezanih podatkov, temveč je namenjena zdravstvenemu osebjju, ki z njeno uporabo stalno skrbi za zdravstveno stanje posameznika. Zdravstveni zapis vsebuje informacije, ki so celovite, popolne in dostopne. Elektronska različica zapisa (EHR – Electronic Health Record) vse to ohranja in dodatno nudi zdravnikom kar največjo stopnjo samostojnosti pri njihovem delu, kakor tudi učinkovito protokoliranje informacij med zdravniki in pacienti, samimi zdravniki in ostalim osebjem.

Anamneza, diagnoza in zdravljenje v kartotečni rubriki nastopajo skupaj. Če njihovo vpisovanje traja v povprečju 15-20 sekund, računalniški sistem za vpis enake vsebine zahteva 1-2 minuti časa. Uporaba elektronskega zapisa pomeni spopad s sodobnimi izzivi pri podpori procesom kliničnega zdravljenja, ki zahtevajo visoko stopnjo obravnave informacij in domenskega znanja ter uporabo bogatega besedišča za izražanje raznolikosti in kompleksnosti bolnikove bolezenske problematike. Za učinkovito izražanje kliničnih vsebin potrebuje sistem EHR skupno terminologijo ter strukturirano in/ali pol-strukturirano interpretacijo vsakega vnosa informacij v sistem.

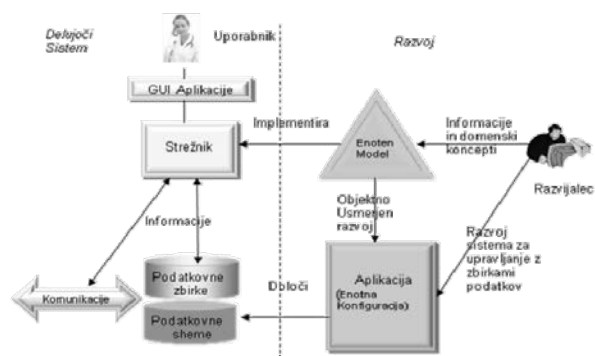
Način, kako so klinične izjave hierarhično ugnezdene v evidenco, daje pomemben okvir za njihovo razlago. EHR sistem omogoča grupiranje in združevanje izjav pod naslovi in postavkami in visoko stopnjo varnosti hranjenja in uporabe informacij. Tako se ne more zgoditi, da bi se npr. bolnika z družinsko anamnezo sladkorne bolezni, pri katerem je sladkorna bolezen preverjeno izključena, vpisalo v podatkovno zbirko bolnikov s sladkorno boleznijo.

Za zdravnike je vedno bolj pomembno dokumentiranje strokovnih odločitev in izmenjava informacij s kolegi, zato morajo biti zapisi sodnomoedicinsko sprejemljivi, resnicoljubni, omogočati strogo sledljivost avtorstva in vnašanja sprememb.<sup>1</sup> EHR zapisi morajo biti narejeni v roku, ki zagotavlja, da lahko osebe, ki so povezane

z nego bolnikov, lahko urejajo svojo evidenco za brezhibno integracijo v celoten sistem EHR.<sup>2</sup> Sodelovanje v sistem EHR mora biti omogočeno tudi ostalim negovalnim organizacijam (domovom za starostnike), študentom medicine in bolnikom samim (vpisujejo vtise o zdravstvenem stanju, potrebe).

## Enonivojska metodologija

Vsak informacijski sistem (IS) je namenjen ustvarjanju in procesiranju tistih konceptov, ki jih sistem razume (npr. oseba, naročilo, diagnoza, zdravlilo), pri čemer so koncepti strukturirani v vedenjskem smislu. Koncepti so skladni opisi domenskih pojmov. Opredeljeni so ločeno po področjih uporabe in so namenjeni posredovanju informacij. O konceptih govorimo takrat, ko jih uporabniki prepoznajo, so označeni (imajo edinstveno ime), samostojni in tvorijo različne vrste informacij, ki so namenjene izmenjavi. Ker se koncepti pogosto pojavljajo v obliki semantične mreže dejstev, klasifikacij, definicij, je potrebno uporabljati jezik, ki omogoča človeku in računalniku učinkovito procesiranje domenskega znanja<sup>3</sup> kot zbirke konceptov. Informacijski sistem, ki je zgrajen po enonivojski metodologiji, prikazuje slika 1.



**Slika 1** Informacijski sistem po enonivojski metodologiji.

Primerki konceptov (ang. Instances of the Concepts) se v informacijskem sistemu shranjujejo in preoblikujejo v človeku razumljivo obliko.

Dandanes podatkovne sheme in zbirke, programska oprema in grafični uporabniški vmesnik temeljijo na objektno usmerjenih (OO) informacijskih tehnologijah in entitetno-relacijskih (ER) modelih. Pri tem so koncepti kodirani v relacijske sheme in s tem formalno v programski kod. V objektno usmerjenih sistemih in modelih je semantika celotnega sistema opisana z jezikom UML (ang. Unified Modeling Language<sup>4</sup>). Mnogi sodobni informacijski sistemi v fazi modeliranja uporabljajo objektno usmerjene modele, a so potem prevedeni in delujejo na osnovi relacijskih podatkovnih shem. Takšna kombinacija potem često privede do težav v aplikacijski programski opremi.<sup>5</sup>

### **Pomanjkljivosti enonivojske metodologije**

V enonivojskih modelih informacijskih sistemov so realizirane samo tiste zahteve, ki so definirane v trenutnem razvoju programske opreme. Model vsebuje hkrati splošne (npr. demografski podatki o pacientu) in domenske koncepte (npr. krvni tlak, sladkorna bolezen). Če se vsi ti koncepti nahajajo v isti dednostni hierarhiji,<sup>6</sup> je lahko mešanje dednih lastnosti (generične lastnosti na višji hierarhiji in specifične lastnosti na nižji) problematično in otežuje večkratno uporabo istega programskega koda. Programske težave, kot je "krhkost objektnih razredov",<sup>7</sup> je potrebno zaznati in preprečevati takoj na začetku programiranja sistema in ne kasneje pri vzdrževanju programske opreme, kar je pri enonivojski metodologiji težko. Število domenskih konceptov v ontologiji<sup>8</sup> je lahko zelo veliko (npr. v medicini<sup>9</sup>), ali pa jih je težko razvrstiti v razrede, zaradi česar je težko izpolniti vse zahteve modela.

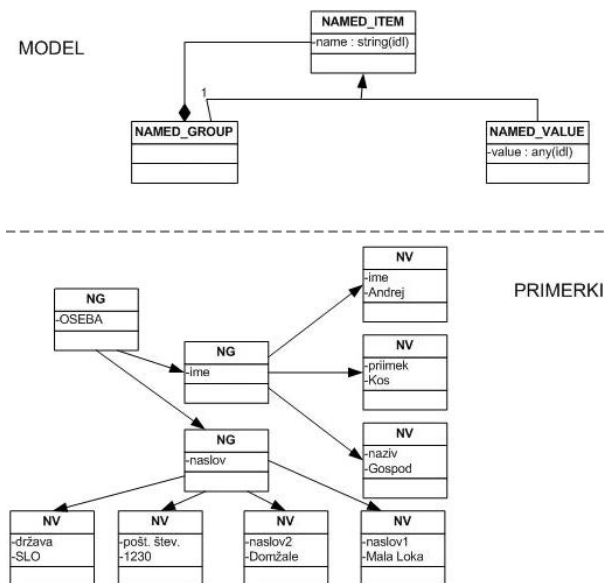
Pri modeliranju sistema sodelujeta dve vrsti strokovnjakov: tisti z bogatim domenskim znanjem in razvijalci programske opreme. Prvi so prisiljeni izraziti koncepte z uporabo programskih jezikov (npr. UML), razvijalci pa imajo pogosto težave z razumevanjem številnih strokovnih pojmov. Posledica je slabše modeliranje procesov, v katerih so domenski (strokovni) pojmi pogosto izgubljeni ali nepravilno izraženi v aplikativni programski

opremi (APO). Uvajanje novih konceptov zahteva poleg spremembe aplikativne programske opreme tudi konverzijo obstoječih podatkov, kar je povezano z velikimi stroški in zakasnitvami pri uvajanju programske opreme v produkcijo. Interoperabilnost<sup>10</sup> je težko doseči. Nenehno je potrebno skrbeti za združljivost različne programske opreme, kar se doseže preko številnih programskih vmesnikov (API). API vnašajo v sistem dodatno nezanesljivost in varnostne težave. Tudi ko je na neki ravni interoperabilnost sistemov dosežena, se ta na splošno s časoma poslabša zaradi odstopanj od dogovorjenih skupnih modelov. Pri sistemih, kjer gre za modeliranje velikega domenskega znanja, je logistično in tehnično (ter včasih tudi politično) težko doseči dogovor o skupnem modelu med različnimi ponudniki in uporabniki. Pomanjkanje standardizacije povzroča težave v interoperabilnosti sistemov in avtomatskem procesiranju (podpora odločanju,<sup>11</sup> rudarjenje s podatki<sup>12</sup>). Glavni razlog, zakaj izgradnja enovitega informacijskega modela ne prinaša dolgoročnih rešitev, pa je v tem, da so informacije in domensko znanje zlit skupaj.

Poglejmo primer. Pacientka ima v mirovanju krvni tlak 110/80 (to je informacija). Krvni tlak je koncept, ki vsebuje dva kvantitativna podatka: sistolični in diastolični tlak, izražen v mmHg. Ob tem se ponavadi podaja še protokol merjenja: pacientov položaj (sedeč, ležeč, stoječ), vrsta merilnika, manšeta in njena namestitvev (vse to je znanje). Ob tem vidimo, da so izjave, ki so nosilke znanja, pripete na razmeroma nestanovitne informacijske entitete, s čemer postane tudi znanje nestanovitno. Objektni model tistega dela informacijskega sistema, ki se nanaša na medicinsko znanje, vsebuje ponavadi veliko količino atributov in vrednosti, ki v nadaljnjih analizah zahtevajo navezovanje na dodatne referenčne tabele, funkcije in kontrole različic. To zahteva še velike količine vnosov podatkov v sistem, za katerim slonijo kompleksni podatkovni modeli.

## Dvonivojsko modeliranje

V enonivojsko modeliranem sistemu dolgoročno z veliko verjetnostjo pride do težav, ker so domenski koncepti neposredno kodirani v formaliziranih programski opre in podatkovnih baz. Včasih se lahko tem težavam izognemo tako, da uporabljamo programske jezike (npr. Java), ki omogočajo dinamično dodajanje novih razredov, s katerimi ublažimo vpliv nestanovitnih podatkov na domenske koncepte. V tem primeru gre za izgradnjo programske opreme, ki temelji samo na modelu informacijskih konceptov. Na primeru (slika 2) postanejo koncepti *ime*, *naslov*, *spol* in *datum rojstva* neustrezni, če se kateri od teh podatkovnih tipov spremeni (npr. *priimek*). Zato nekaterim podatkovnim tipom v modelu dodamo t.i. etiketirane objekte. V našem primeru konceptoma *ime* in *naslov* dodamo razred NV (*named\_value*), s čemer smo konceptoma omogočili spreminjanje, čeprav s tem nekoliko okrnemo referenciranje v podatkovnih zbirkah.



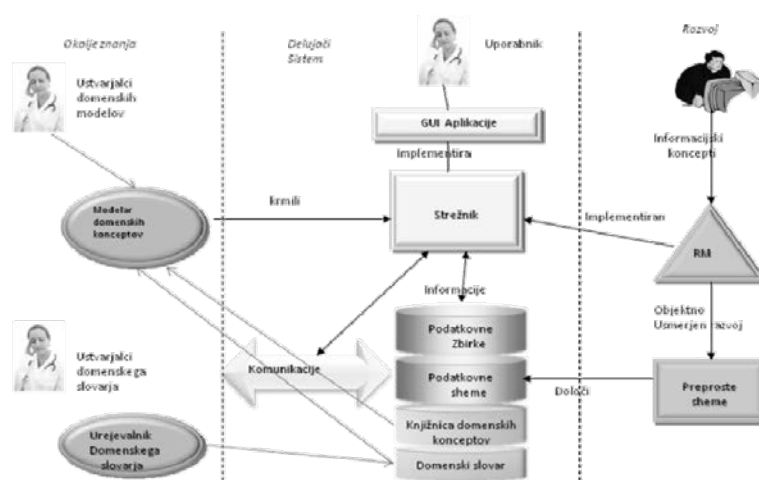
**Slika 2** Generično modeliranje znanja.

Za informacijske sisteme v zdravstvu se vedno bolj zahteva, da so usmerjeni v znanje, kar pomeni, da je konkretna semantika vedno manj prisotna v konkretnem informacijskem modelu. Sistem mora

biti sposoben interpretirati značilnosti, vrednosti, strukture, omejitve in poslovna pravila posameznih konceptov. To je možno doseči s t.i. hierarhijo etiketiranih objektov, kjer razredom NV v modelu dodamo še razrede NG (*named\_group*), ki opisujejo sestavo razredov NV. Z ustvarjanjem hierarhije konceptov, znanje o njih opisujemo (modeliramo) preko konfiguracijskih datotek in preko novih razredov v modelu, v katerih so objekti konkretnega domenskega znanja.<sup>13</sup> Takšno procesiranje domenskega znanja je možno le ob uporabi skupne (dogovorjene) terminologije. Ker zahtevamo od informacijskih sistemov v zdravstvu tudi visoko stopnjo medsebojnega delovanja, sta skupna terminologija in generični referenčni modeli osnovi za delovanje teh sistemov.

Ločitev informacijske semantike in semantike znanja prikazuje slika 3. Domenski specialisti so s tem pri svojem delu postali povsem samostojni in ne potrebujejo informacijskih znanj ali podpore informatikov. Na razvojno-tehničnem delu imamo referenčni model (RM), v katerem so kodirani osnovni in stanovitni koncepti. Vsi ostali (razširjeni) koncepti in formalizmi domenskega znanja (npr. *krvni tlak*, *vrsta poroda*, *otrokova teža ob rojstvu*, *ocena APGAR* ipd.) pa so opisani in shranjeni v konceptnih knjižnicah (drugi nivo modeliranja). Za modeliranje, upravljanje in vzdrževanje teh meta-podatkov je potrebna posebna programska oprema. Razvoj in kasnejša nadgradnja sistema pa je odvisna zgolj od referenčnega modela. Novi domenski koncepti se vključujejo v sistem, ne da bi bilo pri tem potrebno spreminjati aplikativno programsko opremo sistema.

Prednosti dvonivojskega modeliranja je veliko. Izgradnja programske opreme in zbirke podatkov so odvisni samo od razmeroma majhnega števila nespremenljivih informacijskih modelov, zato sta razvoj in implementacija sistema hitrejša. Ker v programski opre ni vgrajenega domenskega znanja, se programska oprema praktično ne spreminja in enostavno jo je nadgrajevati.



**Slika 3** Dvonivojski sistem za procesiranje znanja.

Tehnični model sistema razvijajo informatiki in programerji, medtem ko koncepte znanja definirajo in razvijajo ljudje, ki o njih vedo največ – domenski specialisti gradijo artefakte v realnem času, s katerimi upravljajo delovanje celotnega informacijskega sistema. Interoperabilnost z ostalimi informacijskimi sistemi je zagotovljena hkrati na podatkovnem in konceptualnem nivoju. Z vsem tem so dani pogoji za učinkovito shranjevanje podatkov, informacij in znanja v dokumentno usmerjene zbirke.<sup>14,15</sup> Tako bi na primer morali samo za shranjevanje zapisa o zdravstvenem stanju pacienta (slika 4) v objektivno-relacijske podatkovne zbirke<sup>16</sup> zgraditi objektivno usmerjen model z najmanj petimi tabelami. Še bolj učinkovit sistem povpraševanja po informacijah, predvsem pa po domenskem znanju, dosežemo z integracijo grafično usmerjenih podatkovnih zbirk<sup>17,18</sup> v dvonivojski sistem.

## Referenčni model

Referenčni model (RM) ali informacijski model sistema EHR vsebuje tiste koncepte, ki so večinoma prisotni na začetku kliničnega zdravljenja. Gre za koncepte, ki so vezani na zagotovljeno zdravstveno varstvo pacientov, predhodne preiskave, laboratorijske teste ipd. Izjave, manjša poročila in navedbe, ki so v obliki besedil, slik ali audio-vizualnih datotek, lahko

zdravniki-specialisti pred shranjevanjem v sistem opremijo z atributi časa, kraja, vrste in lastništva. Pri tem so nekateri koncepti in pripadajoči atributi stanovitni, večina pa jih je časovno spremenljivih. V sistem EHR se ti koncepti vnašajo v t.i. protokolarno okolje, ki je razdeljeno po smiselni odsekih (slika 5).

Kompozicija ali transakcijski koncept je znotraj protokolarnega okolja zasnovan tako, da ne zagotavlja le osnovnih podatkov o pacientu, ampak vse potrebne informacije zato, da lahko klinično zdravljenje poteka učinkovito in uspešno. Znotraj tega koncepta tečeta najmanj dve vrsti informacijskih tokov. Za dogodke (kot so npr. obisk pri zdravniku, rezultati testiranja, radiološke preiskave ali kirurški poseg) je značilno, da so vezani na stroške, ki jih plačujejo pacienti ali zdravstveno zavarovanje. Dogodek poleg vsebine nosi s seboj attribute časa, kraja in udeležencev. Trajne postavke lahko bistveno vplivajo na potek kliničnega zdravljenja in so razvrščene po kategorijah, kot so družinske/dedne bolezni, socialno stanje, jemanje zdravil, predpisane terapije, načrt zdravljenja in življenjski slog.

Kompozicija kot ključni koncept referenčnega modela EHR izpolnjuje sledeče kriterije: zagotavlja shranjevanje in dostopnost zapisov o zdravstvenem stanju pacienta; varnost in konsistentnost je zagotovljena za vsak, še tako skromen podatek; če več uporabnikov istočasno uporablja sistem, ne



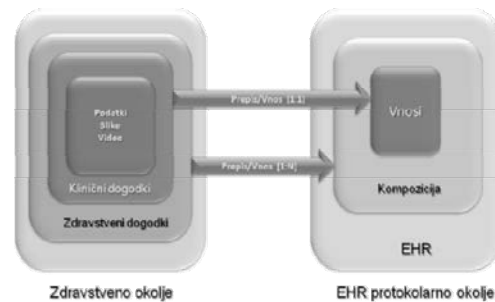
prihaja do medsebojnega oviranja; za vsak podatek, vnešen v sistem, je zagotovljeno, da bo vsak izbris zaznamovan z izvajalcem izbrisa; uporabniki sistema imajo možnost spreminjati in posodabljeni vsebine EHR ter odpravljati napake; za vsak podatek, ki se vnese v sistem, je zagotovljena sledljivost.

```

"_id" : ObjectId("4c537d4b82fd211170000000");
"ime" : "Andrej Novak";
"datum_tvorbe_zapisa" : "Torek Sep 2011 11:32:58 CET";
"Prvi_obisk" : "Pet Jan 10 1970 00:07:01 CET";
"plačnik" : {
  "description" : "ZZZS",
  "id_nosilca" : "0125005",
  "OK" : "Ljubljana",
  "id_osebe" : "030007746",
  "roj_datum" : "6.11.1951",
  "državljanstvo" : "SLO"
};
"bivanje" : {
  "opis" : "stanovanjska hiša",
  "telefon" : "015628270",
  "naslov" : "Mala Loka 27",
  "naslov_2" : "",
  "mesto" : "Domžale",
  "pokrajina" : "",
  "poštna_številka" : "1230",
  "država" : "SLO"
};
"zdravila" : {
  "_id" : "4c537b3382fd21df6f040000",
  "ime" : "Bloxan",
  "opis" : "Zaviralec beta",
  "odmerek" : "50mg dvakrat/dan",
  "začetek_jemanja" : "Pet 30. Julij 2010 21:32:58 CET"
};
{
  "_id" : "3e53745382fd21ci6f0872530",
  "ime" : "Asentra",
  "opis" : "Selektiven zaviralec ponovnega privzema serotonina",
  "odmerek" : "100mg enkrat/dan",
  "začetek_jemanja" : "Pet 30. Julij 2010 21:32:58 CET"
};
"naslednji_obiski" : {
  "datum" : "Sre 29. Feb 2012 09:15:00 CET",
  "povzetek" : "Stanje glede na srčno aritmijo in depresivne motnje"
};
"specialistični_pregledi" : {
  "datum" : "Sre 10. Avg 2011 11:00:00 CET",
  "ustanova" : "UKC-Klinika Petra Držaja, oddelek za hipertenzijo",
  "pregledi" : "24urno merjenje utripa in krvnega tlaka"
};
"stanje" : {
  "anamneza" : "Srčna aritmija",
  "Depresivne motnje"
}

```

**Slika 4** Primer kartotečnega vpisa.

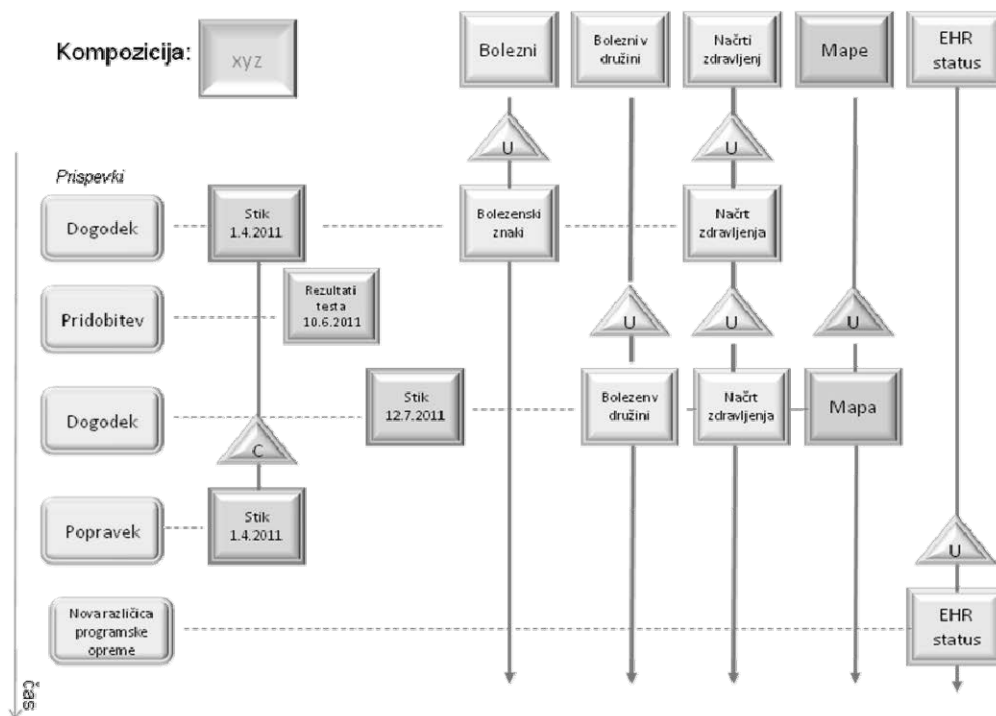


**Slika 5** EHR protokolarno okolje.

EHR statusi so tiste informacije, ki so potrebne za učinkovito upravljanje informacijskih tokov. Sistemu EHR povedo, ali je določen informacijski tok (zaporedje dogodkov in postavk) v stanju uporabe, spreminjanja, dopolnjevanja ali brisanja. EHR statusi so neposredno povezani z vzajemnim delovanjem agenta (npr. računalniški program) na sistem EHR. Tako se pri vnosu vsakega podatka v sistem beleži kdo, kdaj in kje je vnos opravil. Še več, sistem EHR sproti gradi učinkovito strukturo map (znotraj kompozicije), kjer se podatki ali informacije shranjujejo po tematskih sklopih.

Podsistem za upravljanje sprememb (CM) v sistemu EHR omogoča, da lahko kadarkoli pridobimo vse, kar je bilo zapisano v sistem. To je še posebej pomembno takrat, ko je potrebno opraviti izredne analize kliničnih procesov z medicinsko-pravnega vidika.<sup>19,20</sup>

Na sliki 6 je prikazan primer kompozicije, ki jo sestavlja pet prispevkov. Prvi prispevek je vezan na pacientov obisk pri zdravniku. Kot dogodek se zabeleži stik z zdravnikom, spremembe na listi bolezenskih težav, seznamu zdravil in načrtu zdravljenja. Naslednji prispevek je vezan na pridobitev rezultatov analiz iz laboratorija. Pri ponovnem obisku pri zdravniku se obnovijo informacije o družinskih in dednih boleznih in dopolni načrt zdravljenja pacienta. Četrty prispevek h kompoziciji je namenjen odpravljanju napak v zdravstvenih zapisih (npr. napačno črkovan priimek pacienta). Zadnji prispevek je vezan na namestitev nove različice programa. Za vsak zapis znotraj kompozicije se v sistemu EHR beleži njegova različica, in sicer tako, da imajo vsi zapisi, ki pripadajo istemu prispevku, isto različico.



Slika 6 Primer kompozicije.

## Arhetipni model

Vsak koncept ima lahko številne različice svoje osnovne opredelitve. Koncept *oseba* na primer ustreza veliko različicam koncepta *človek*. Vključuje koncepte, kot so zunanji videz, osebnost, vedenje, ipd. Vprašajmo se, katera je tista stopnja spremenljivosti, ki določenega koncepta še ne spremeni do te mere, da ta postane povsem drugačen? Odgovor na to vprašanje dobimo tako, da v model vpeljemo t.i. *strukturne omejitve konceptov* (ang. Structural Constraint Definition) – množice omejitev, ki omejujejo število različic istega koncepta. S tem ti postanejo mnogo bolj definirani, kot tisti v modelih, ki so zasnovani na predlogah. Cilj torej ni operirati z enim dovršenim konceptom (glej primer koncepta krvnega tlaka), temveč dati uporabnikom možnost, da uporabljajo številne različice istega koncepta. Ta možnost je v sistemu EHR zagotovljena tako, da so koncepti omejeni po strukturah, vrstah, vrednostih in obnašanju. Domenski koncepti so s tem formalno oblikovani

kot arhetipi, ki jih nato uporablja sistem v realnem času.

Arhetipi<sup>21</sup> so omejeni gradniki domenskih konceptov. Imenujemo jih tudi strukturirana poslovna pravila, ki opisujejo zgradbo informacijskih primerkov, katerih pripadajoči razredi so opisani v referenčnem modelu. Analogijo za arhetipe najdemo v LEGO kockah, če si predstavljamo, da referenčni model podaja specifikacijo LEGO kock, s katerimi potem ustvarjamo različne konstrukcije (hiša, traktor, avto ipd.). Semantika RM je torej analogna semantiki LEGO kock, kjer je definirana vrsta in velikosti konstrukcije (tj. velikost informacijskega prostora) oziroma omejitvam arhetipov v arhetipnem modelu.

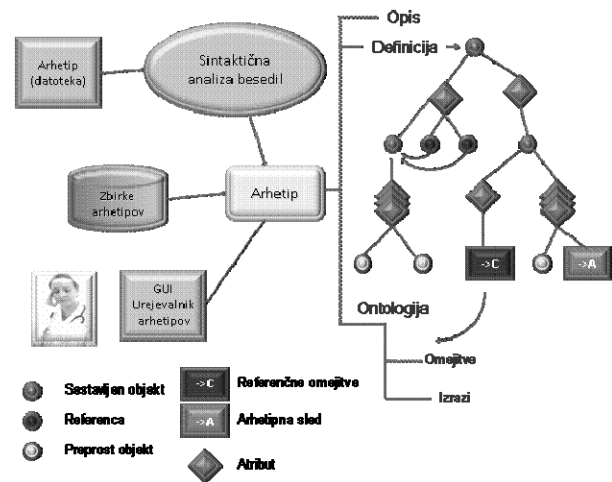
Arhetipni model<sup>22,23</sup> (AM) je prototipni model, ki omogoča uporabo številnih različic konceptov. Definira strukture in semantiko arhetipov in predlog. Sestavljajo ga *arhetipni opisni jezik* (ADL – Archetype Description Language), *arhetipni objektni model* (AOM), *arhetipni profiler* (OAP) in

*objektni model predlog* (TOM). Vse naštetu so pristni objektno usmerjeni modeli. Medtem, ko ADL skrbi za sintakso tekstualnih arhetipov in predlog, AOM kot generični model izraža arhetipe za katerikoli referenčni model. ADL in AOM delujeta skupaj v *sintaktični analizi besedil* (ang. Parser<sup>24</sup>), orodju, ki iz besedila v obliki ADL generira objekte, primerne za shranjevanje v podatkovne zbirke. Podobno TOM določa objektni model predlog in s tem zagotavlja združevanje arhetipov v skladu z ekranskimi obrazci v grafičnem uporabniškem vmesniku (GUI – Graphical User Interface).

Na sliki 7 so prikazane različne metode, s katerimi ustvarjamo arhetipne objektne strukture. Rezultat sintaktične analize arhetipov, ki so predstavljeni v obliki zapisov XML, OWL ali ADL, so strukturirani objekti (grafi). Uporabniki ustvarjajo arhetipe z uporabo grafičnega urejevalnika arhetipov in jih shranjujejo v podatkovne zbirke. Definijski del arhetipa je objektna drevesna struktura, ki jo sestavljajo vozlišča in povezave med njimi. Vozlišča predstavljajo preproste ali sestavljene objekte (lastnosti in omejitve konceptov), referenčne objekte (kažejo na druge objekte), arhetipne sledi (izjave o lastnostih in omejitvah arhetipov, na osnovi katerih nastanejo novi arhetipi; semantika teh objektov je praktično enaka sestavljenim objektom originalnega arhetipa), enojni in mnogokratni atributi (lastnosti in omejitve objektov) ter referenčne omejitve (referenca lastnosti in omejitev na arhetip v eksterni ontologiji ali terminologiji arhetipov preko jezika ADL).

V znanje usmerjene metodologije omogočajo, da v informacijskem sistemu ločimo opisovanje konceptov od tehničnega modela samega sistema. Želimo najti način, kako interpretirati številne kompleksnosti, ki jih tvorijo domene v realnem svetu, in sicer tako, da so razumljive človeku in računalniku. Če si domensko znanje (npr. iz medicine ali mikrobiologije) predstavljamo v večrazsežnem prostoru, so koncepti razvrščeni na različnih področjih oziroma ravneh tega prostora. Zbir konceptov na vsaki ravni predstavlja ontologijo te ravni. Koncepti na zunanji ravni so

na splošno sestavljeni iz tistih na notranji ravni z vsemi njihovimi lastnostmi, ki jih od njih dedujejo.



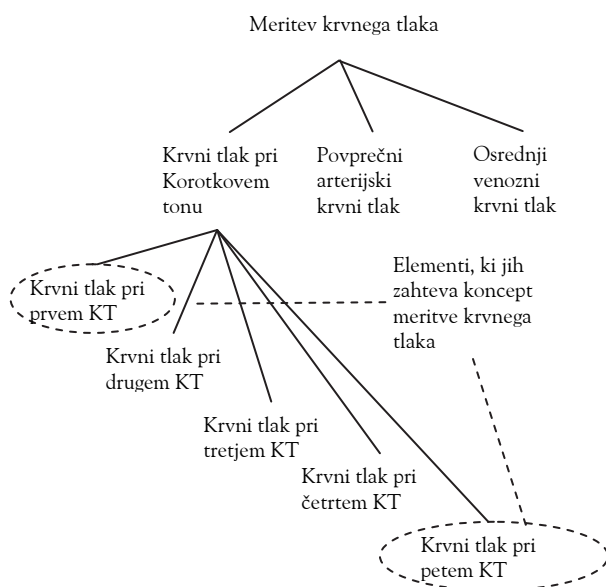
**Slika 7** Modeliranje arhetipov.

Da je domensko znanje računalniku razumljivo, ga je potrebno modelirati. *Ontologija* je tista, ki na ravni računalniškega sistema formalizira domensko znanje.<sup>8</sup> Formalizacija domenskega znanja poteka na najmanj treh ravneh. Prvo raven (raven 0) tvorijo domenski jeziki in načela. V medicini se načela nanašajo na teme, kot so anatomija, parazitologija in farmakologija. Sledijo znanja o medicinskih procesih (npr. razvoj zarodka) in splošno sprejetih dejstvih (npr. delovanje srca). Ta raven znanja je stanovitna, ni odvisna od uporabnikov (npr. v zdravstvu ali izobraževanju) in znanje večinoma najdemo v učbenikih, znanstvenih člankih in usposabljanjih. Pravimo, da uporabniki nimajo izoblikovanih svojih stališč do teh znanj. Koncepti, ki zahtevajo posebne okoliščine uporabe in so negotovi ali odvisni od drugih ontologij, na to domensko raven ne sodijo.

Medicina je ena od redkih znanstvenih področij, ki ima domensko znanje tako rekoč pripravljeno za računalniško procesiranje.<sup>25,26</sup> Tako obstajajo ontologije v medicini (npr. SNOMED CT<sup>27</sup>), ki opisujejo osnovne pojme in podajajo množice semantičnih razmerij, ki so nujno potrebne za računalniško procesiranje znanja v medicini.



Druga raven (raven 1) so vsebine. Na tej ravni postaja znanje vse bolj specifično glede na namen in uporabnike, kar pomeni, da lahko to raven razdelimo na številne podravni. Zaradi tega lahko pojme opredeljujemo ločeno in jim pripisujemo različne omejitve njihovih struktur. Številne stopnje ravni 1 natančneje opredeljujejo pojme. Vzemimo primer medicinske ontologije, kjer so na ničelni ontološki ravni naslednji koncepti: meritev krvnega tlaka, indeks telesne mase in meritve na delu človeškega telesa. Krvni tlak na isti ontološki ravni sestavljata sistolični in diastolični tlak, ki ju odčitamo ob prvem in petem Korotkovem tonu. V ontologiji na ničelni ravni ne bomo našli povezave med konceptoma krvnega tlaka in Korotkovega tona (KT), temveč bo govora o venoznem in arterijskem krvnem tlaku tako, kot je prikazano na sliki 8. Ti dve vrednosti krvnega tlaka nudita dovolj informacij o človekovem krvnem tlaku. Če pa uporabljamo koncept meritve krvnega tlaka preko zaznave Korotkovega tona, je potrebno na drugem ontološkem nivoju razložiti merjenje krvnega tlaka pri KT.



**Slika 8** Ontologija merjenja krvnega tlaka.

Na naslednji ontološki ravni (raven 2) so koncepti organizacije. Uporabniki oblikujejo to raven z namenom učinkovite navigacije med nepovezanimi informacijami, oziroma z

medsebojnim povezovanjem, ki se doseže z uporabo različnih metodologij in procesov. Primer koncepta, ki sodi v tretjo ontološko raven, je problemsko naravnani zdravstveni zapis (SOAP<sup>28</sup>), ki ga po navadi zdravniki družinske medicine oblikujejo v hierarhično strukturo oziroma zaporedje načrtovanih aktivnosti (hospitalizacija, laboratorijske analize, večdnevna dieta, odpust iz bolnišnice, opazovanje na domu) oziroma pretoka informacij in standardizirajo.

Raven 3 opisuje koncepte, povezane s shranjevanjem informacij. Če smo na prejšnjih ravneh ustvarili in primerno uredili vsebine (izražene z osnovnimi elementi in besednjakom), na tej ravni te vsebine smiselno povezujemo v informacije ali znanje. Zdravstveni karton vsebuje vse informacije o pacientovem zdravstvenem stanju. Informacije morajo biti na tej ravni ne le same po sebi smiselne, temveč tudi jasne iz samega konteksta. Vsebovati morajo vse podatke v zvezi z zbiranjem in oblikovanjem (npr. identiteta zapisovalca, datum/čas zbiranja). Koncepti na ravni 3 so razvejani in vključujejo npr. družinsko anamnezo, zdravila, ki jih pacient jemlje, terapevtske ukrepe in cepljenja.

Zadnja ontološka raven se nanaša na izbiro in pakiranje informacij, namenjenih komunikaciji z drugimi sistemi. Značilni koncepti te ravni so različni dokumenti, poročila, izpiski. EHR izpis<sup>29</sup> je koncept, ki zagotavlja komuniciranje sistema EHR z drugimi informacijskimi sistemi in je del referenčnega modela sistema EHR. Zahtevek za komunikacijo se posreduje v obliki elektronskega sporočila ali specifične medmrežne storitve in vsebuje podrobnejšo specifikacijo vsebine odlagališča informacij, ki se izpiše in posreduje drugemu sistemu. V odgovoru je ponavadi vsebina strukturirana tako, kot določajo priporočila CEN EN13606.<sup>30</sup> Informacijski model *openEHR ekstrakt* je popolnoma nevtralen glede na različne komunikacijske tehnologije med sistemi. Angleščina je "lingua franca" sporazumevanja v tem sistemu. Tabela 1 povzema nivojsko klasifikacijo znanja sodobnega informacijskega sistema za podporo procesom kliničnega zdravljenja.

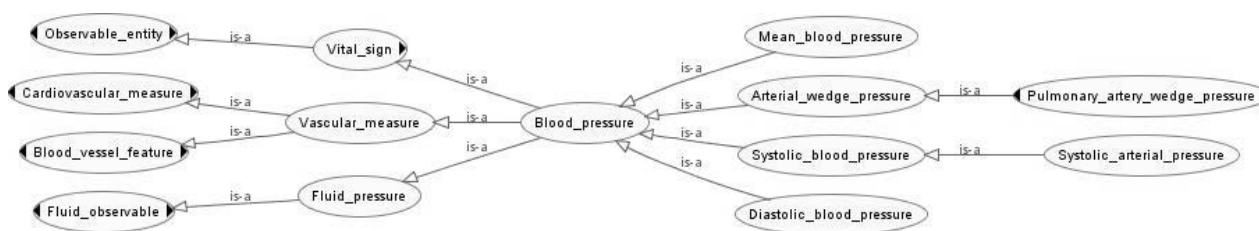
**Tabela 1** Klasifikacija znanja v kliničnem informacijskem sistemu.

Raven	Pomen	Razlaga	Primer
0 Načela	Besedišče in ostala semantika domene; dejstva, ki veljajo za vse primere in vse kontekste uporabe.	Semantične mreže, nadzorovan besednjak	- Učbeniki - Ontologija SNOMED
1a Vsebine (povsodne)	Uporablja kontekstno odvisne koncepte, ki jih razumejo vsi domenski uporabniki.	V sestavi ravni 0	- Krvni tlak - Napotnica za spec. pregled
1b Vsebine (na osnovi primerov)	Kontekstualno odvisni koncepti, določeni glede na posebne primere uporabe.	V sestavi ravni 0	- Neželeni učinki - Družinska anamneza
2 Organizacijska	Strukturni koncepti, katerih namen je organizirati podatke na enak način, kot ti nastopajo v pisnih dokumentih.	Hierarhična struktura konceptov na ravni 1	- SOAP zdravstveni zapis - Kajenje in uživanje alkohola - Družinska anamneza
3 Shranjevanje	Koncepti, ki se nanašajo na strukturiranje informacij za shranjevanje.	Koncepti na ravni 2	- Recepti za zdravila - Seznam zdr. težav - EHR
4 Komunikacija	Koncepti v zvezi s pakiranjem informacij za izmenjavo.	Koncepti, ki izhajajo iz ravni 3	- Različni dokumenti - EHR povzetki

## Mobilizacija znanja

Osnovni namen arhetipnega pristopa je omogočiti zdravnikom, da gradijo, nadgrajujejo in uporabljajo znanje, ki ga vsebujejo arhetipi. Večino osnovnih arhetipov navadno zgradi majhna skupina ekspertov klinične medicine (EKM). To pomeni, da svoje neotipljivo znanje preoblikujejo v arhetipno strukturo z uporabo urejevalnika arhetipov – računalniškega programa z grafičnim uporabniškim vmesnikom.<sup>31</sup> Ustvarjanje arhetipov ne sodi v vsakodnevno zdravnikovo delo, ampak so narejeni arhetipi podlaga za vnašanje podatkov in informacij o poteku kliničnega zdravljenja. Ob tem EKM prevzamejo odgovornost za nadaljnje upravljanje in širjenje arhetipiranega znanja. Ta korak je potreben zaradi številnih vzrokov: vsebina arhetipa mora biti dokazljiva, vsebuje naj dobre prakse in čim več neotipljivega in s konsenzom usklajenega znanja; glede na nova znanja je potrebno arhetipe stalno vzdrževati in stremeti k njihovi standardizaciji; v čim večji meri je potrebno izločati prekrivanje arhetipov in omogočati njihovo čim lažjo dostopnost (na medmrežju).

V *openEHR* ontološki strukturi sestavljajo *ontologijo dejstev* trije sklopi: (1) klasifikacija bolezni<sup>32</sup> in klasifikacija osnovnega zdravstva,<sup>33</sup> (2) opis zdravstvenih procesov in (3) ontologije oziroma terminologije (npr. SNOMED CT, UMLS<sup>34</sup>), kjer so koncepti diskretne entitete v domenski ontologiji. Ta formalizacija ne vnaša v celoten sistem prevelikih omejitev, če vemo, da dobro strukturirane ontologije pod ravni 0 vsebujejo razmeroma majhno število referenc kateregakoli koncepta do drugih konceptov. Vzemimo kot primer koncept *krvnega tlaka*. Omenili smo že, da ta koncept vsebinsko sodi v prvo raven ontologije medicine in ga razumemo kot skupek dveh količin: sistoličnega in diastoličnega krvnega tlaka. Definicija tega koncepta je jasna in nedvoumna. Primerjajmo ta koncept s konceptom *sistoličnega krvnega tlaka*. Tudi ta pojem je v medicini popolnoma jasen, vendar bi bilo narobe, če bi v informacijskem sistemu beležili ta koncept ločeno od diastoličnega tlaka ali celo brez njega. Oba koncepta vselej nastopata skupaj.



**Slika 9** Grafični prikaz koncepta krvnega tlaka v ontologiji SNOMED.

Povezovanje pojmov v kliničnem arhetipnem modelu z različnimi terminologijami poteka preko podsistema MoST,<sup>35</sup> ki preko različnih semantičnih tehnik za vsak koncept poišče ustrezen izraz. Grafično ponazoritev koncepta krvnega tlaka v ontologiji SNOMED CT prikazuje slika 9. Večina orodij za pregledovanje/urejanje ontologij<sup>36</sup> vsebuje tudi grafični pregledovalnik, ki omogoča učinkovito vizualno predstavitev konceptov znotraj ontoloških razredov in njihovih medsebojnih odnosov. V informacijskem smislu model na sliki 9 ne ponazarja podatkovnih struktur, temveč funkcijo (metodo), ki vselej vrne dve vrednosti krvnega tlaka. V jeziku UML<sup>4</sup> zapišemo to funkcijo kot je prikazano na sliki 10.

Zapis na sliki 10 si lahko predstavljamo kot predlogo (obrazec), v katero uporabnik vpisuje vrednosti, definira strukturo in vsebino koncepta ter te vrne informacijskemu sistemu v nadaljnjo procesiranje. Zdi se, da lahko z objektnim programiranjem učinkovito modeliramo koncepte. Vendar stvar ni tako preprosta, saj zdravniki uporabljajo mnoge različice istih konceptov. Ko govorijo o krvnem tlaku, mislijo na sistolični in/ali diastolični tlak; v pediatriji uporabljajo t.i. "tlak pri 1. in 4. tonu" ali "KT1 in KT4" oziroma "KS1 in KS4". Prav zaradi tega je potrebno v arhetipu/predlogi omogočiti vpisovanje konceptov z različnimi imeni, dodatnimi atributi in pravili. S tem postanejo razmeroma preprosti koncepti bolj kompleksni, kar vodi v njihovo formalizacijo in rušenje skladnosti z značilnostmi pripadajočih ontoloških razredov. Ti problemi se rešujejo preko programskih vmesnikov za izmenjavo znanj.

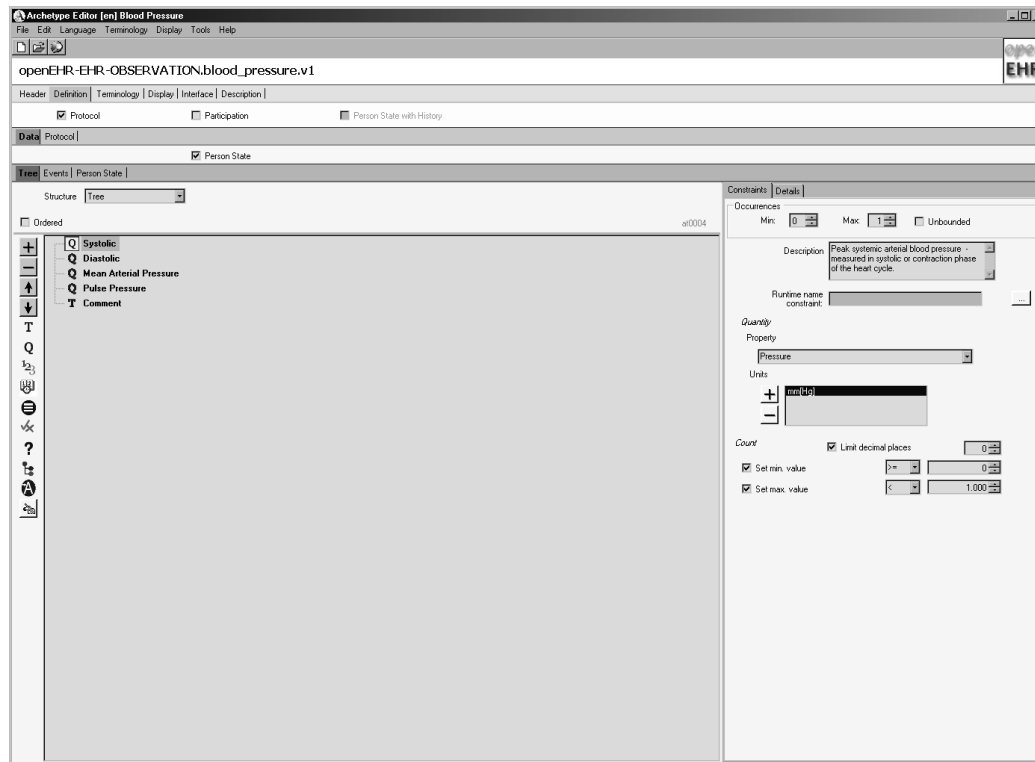
Urejevalnik arhetipov<sup>37</sup> je orodje, ki omogoča zdravnikom ustvarjanje novih arhetipov, učenje iz

že ustvarjenih arhetipov ter upravljanje z njimi. Učinkovito pregledovanje in raziskovanje vsebin je ključnega pomena, zato je urejevalnik podprt z grafičnim vmesnikom (slika 11). Ustvarjene arhetipe zdravniki shranjujejo v lokalne repozitorije ali pa jih nudijo drugim uporabnikom tako, da jih shranjujejo v centralne repozitorije arhetipov na spletu. Avtor arhetipa za koncept *krvni tlak* ga je ponudil ostalim uporabnikom v rabo tako, da ga je shranil na *openEHR* spletni repozitorij.<sup>38</sup> Spletno orodje omogoča uporabnikom, da potrebne arhetipe ali predloge poiščejo, shranijo za lastno uporabo ali vsebinsko preuredijo za lastne potrebe. Arhetipi in predloge so s tem prepuščeni zdravnikom v nadaljnje nadgrajevanje in preurejanje. Novo nastale arhetipe lahko zdravniki in ostali uporabniki preko spletnega repozitorija ponudijo drugim uporabnikom (slika 12). Grafični uporabniški vmesnik pa jim omogoča, da vidijo, kako je domensko znanje o nekem konceptu strukturirano. Primer na sliki 13 prikazuje strukturo domenskega znanja za koncept krvnega tlaka, prevzet iz ontologije SNOMED CT.

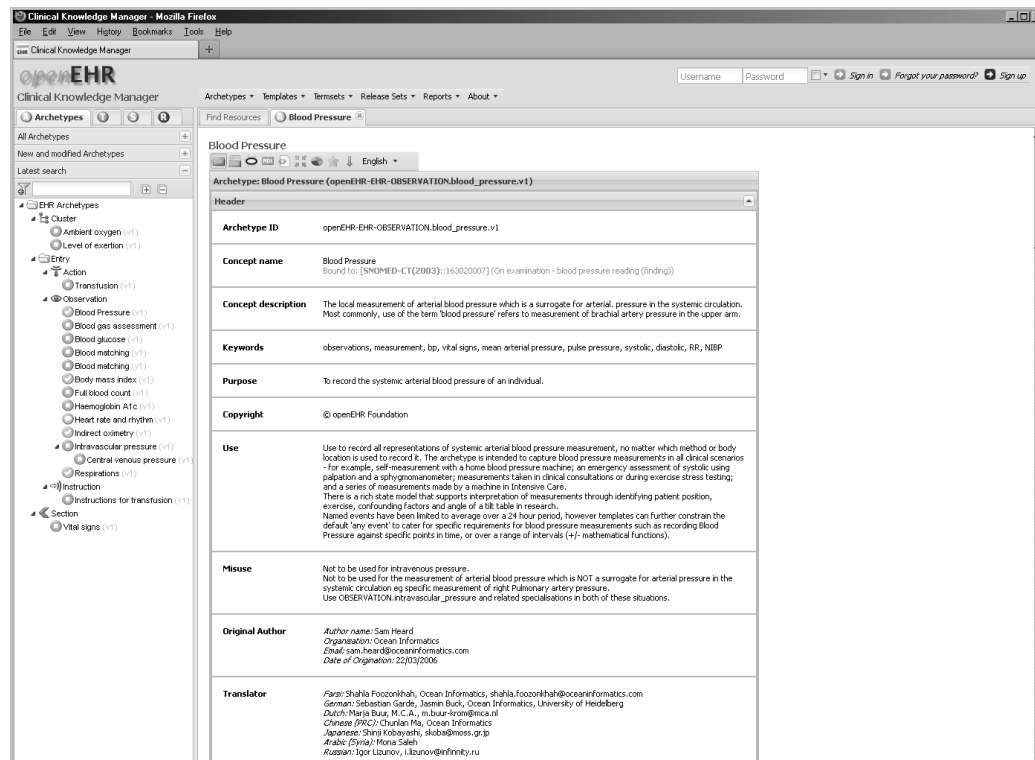
```

class "blood pressure"
feature
  "systolic blood pressure" [SNOMED term nnnnn]: QUANTITY
    ensure: Result.units = UNIT mm [Hg]
  "diastolic blood pressure" [SNOMED term nnnnn]:
QUANTITY
    ensure: Result.units = UNIT mm [Hg]
constraint
  "blood pressure rule" [SNOMED term nnnnn]:
  "diastolic blood pressure".value <= "systolic blood
pressure.value"
end
    
```

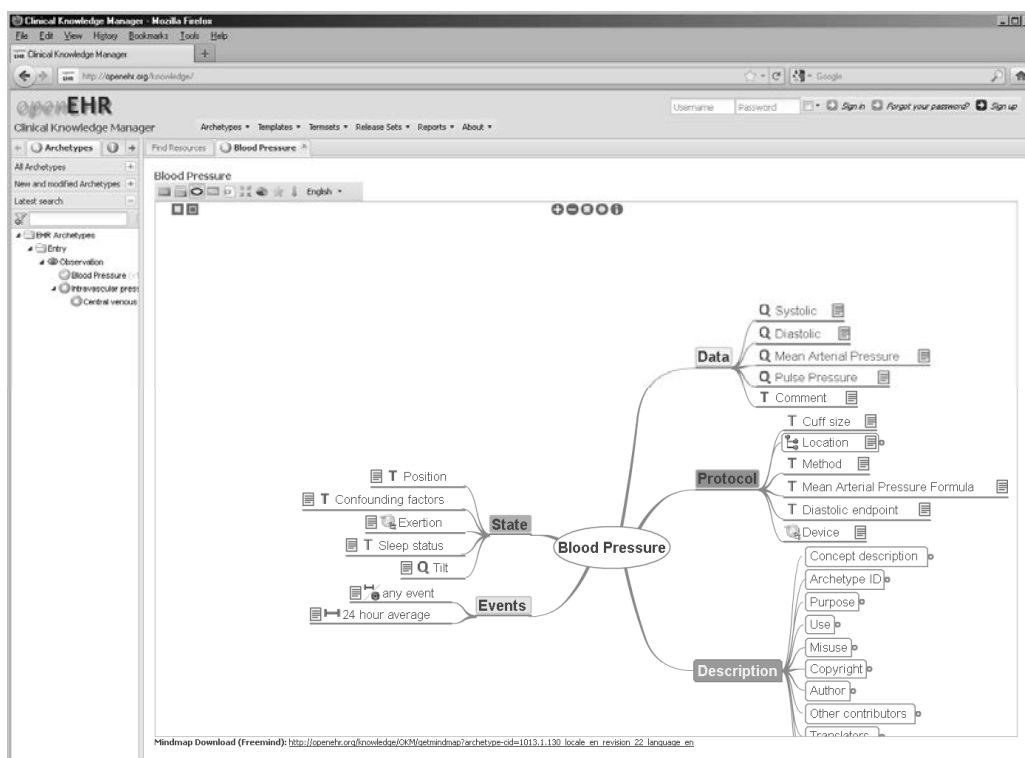
**Slika 10** Funkcija, zapisana v objektnem programskem jeziku.



Slika 11 Grafični uporabniški vmesnik urejevalnika arhetipov.



Slika 12 Osnovni podatki o arhetipu koncepta krvni tlak, ki je shranjen v spletnem repozitoriju.



**Slika 13** Grafični prikaz arhetipa koncepta krvnega tlaka.

## Zaključek

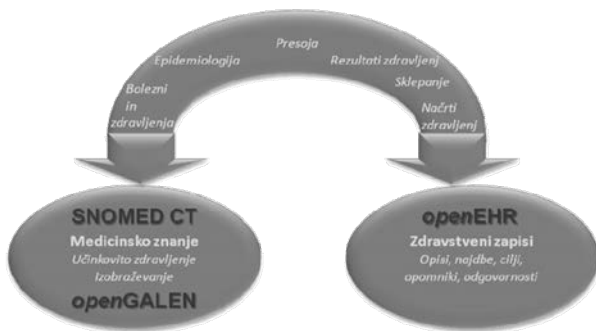
Največkrat je tako, da se poleg začetne cene, zaznane privlačnosti in zanesljivosti dobavitelja (navadno se ta ocenjuje glede na velikost podjetja in promet, ne pa npr. na poznavanje strateških usmeritev v zdravstvu) pri nabavi informacijskega sistema upoštevajo še kriterij, da informacijski sistem nudi dobro podporo delovnim tokovom in sledljivost toku dokumentov ter podporo poročanju in sledenju stroškov. Če je dobavitelj že ali bo zgradil IS po enonivojski metodologiji, potem ta ne zagotavlja skoraj nikakršnega računalniškega procesiranja domenskih znanj, kar se bo krepko poznalo pri kakovosti in učinkovitosti sistema za podporo kliničnim procesom.

Odprihodne rešitve<sup>39,40</sup> v začetku nastajajo znotraj majhnih skupin strokovnjakov, ki jih bolj kot komercialni vodi idejni vzgib. Kasneje se delovanje idejnega prototipa preverja med strokovnjaki-prostovoljci in se z njihovo pomočjo nadgrajuje in spreminja, da bi v nekem trenutku

načrtovan sistem dosegel svojo komercialno zrelost. Glavni cilj sodelovanja v okviru *openEHR* je zgraditi visoko zmogljiv sistem za podporo procesom kliničnega zdravljenja, zasnovanem na zdravstvenem zapisu, medicinskem znanju in protokolih zdravstvene nege (slika 14). Od začetka sodelovanja pri modeliranju sistema sodelujejo tako informatiki kot zdravniki (z različnih področij medicine). Ločitev domenskega znanja od tehnične plati izgradnje informacijskega sistema prinaša izjemne učinke ne le pri vzajemnem delovanju IS v medicini, temveč tudi pri učinkoviti mobilizaciji znanja v procesih v zdravstvu. To potrjuje velik razkorak med številom pojmov (> 1.000.000) in konceptov (> 300.000) v modelu domenskega znanja in referenčnem modelu IS (< 150). Še več, zdravniki in ostalo zdravstveno osebje so s tem pridobili aktivno vlogo pri izgradnji in nenehnemu posodabljanju IS v segmentu krepitve znanj, spoznanj, izkustev in dobrih praks, kar postane uporabnikom dostopno v širšem in ožjem smislu. Skrbno načrtovan in zgrajen referenčni model sodelovanja *openEHR*



zagotavlja veliko prednost tistih IS za podporo procesom kliničnega zdravljenja, ki so zgrajeni na podlagi tega modela, predvsem z vidika dolgoživosti programske opreme. Ta se zelo malo ali sploh ne spreminja, kar prinaša velike prihranke. Ker danes v globalnem smislu ni več ovir na področju informacijsko-komunikacijske infrastrukture (medmrežje), so izpolnjeni vsi pogoji za izpolnitev zahteve po interoperabilnosti kot ene od prednostnih zahtev za vsak IS, še posebej v zdravstvu.



**Slika 14** Informacijske storitve za podporo zdravljenju pacientov.

## Literatura

1. Rector AL, Nowlan WA., Kay S, Goble CA, Howkins TJ: A framework for modelling the electronic medical record. *Meth Inform Med* 1993; 32(2):109-119.
2. Ceusters W: *Referent Tracking: The New Paradigm*. Buffalo 2006: Center of Excellence in Bioinformatics and Life Sciences Ontology Research Group. <http://www.referent-tracking.com/RTU/sendfile/?file=DagstuhlRefTrak.ppt>
3. Wennerberg P, Zillner S, Moeller M, Buitelaar P, Sintek M: KEMM: A Knowledge Engineering Methodology in the Medical Domain. In: Eschenbach C, Grüninger M (Eds.), *Formal Ontology in Information Systems – Proceedings of the Fifth International Conference (FOIS 2008)*. Amsterdam 2008: IOS Press; 79-91. [http://www.osb-i.de/fileadmin/user\\_upload/Presse/Artikel/Zillner\\_KEMMMethodolgy\\_2008.pdf](http://www.osb-i.de/fileadmin/user_upload/Presse/Artikel/Zillner_KEMMMethodolgy_2008.pdf)
4. Unified Modeling Language. In: *Wikipedia, The Free Encyclopedia*. San Francisco 2011: Wikimedia Foundation. [http://en.wikipedia.org/wiki/Unified\\_Modeling\\_Language](http://en.wikipedia.org/wiki/Unified_Modeling_Language)
5. Ambler SW: *The Object-Relational Impedance Mismatch*. <http://www.agiledata.org/essays/impedanceMismatch.html>
6. Inheritance (object-oriented programming). In: *Wikipedia, The Free Encyclopedia*. San Francisco 2011: Wikimedia Foundation. [http://en.wikipedia.org/wiki/Inheritance\\_%28object-oriented\\_programming%29](http://en.wikipedia.org/wiki/Inheritance_%28object-oriented_programming%29)
7. Mikhajlov L, Sekerinski E: A Study of the Fragile Base Class Problem. <http://www.cas.mcmaster.ca/~emil/publications/fragile>
8. Ontology (information science). In: *Wikipedia, The Free Encyclopedia*. San Francisco 2011: Wikimedia Foundation. [http://en.wikipedia.org/wiki/Ontology\\_%28information\\_science%29](http://en.wikipedia.org/wiki/Ontology_%28information_science%29)
9. OBO Foundry. In: *Wikipedia, The Free Encyclopedia*. San Francisco 2011: Wikimedia Foundation. [http://en.wikipedia.org/wiki/OBO\\_Foundry](http://en.wikipedia.org/wiki/OBO_Foundry)
10. Renner SA, Rosenthal AS, Scarano GJ: Data Interoperability: Standardization or Mediation. In: *Proceedings of the First IEEE Metadata Conference*. IEEE Computer Society 1996.
11. Clinical decision support system. In: *Wikipedia, The Free Encyclopedia*. San Francisco 2011: Wikimedia Foundation. [http://en.wikipedia.org/wiki/Clinical\\_decision\\_support\\_system](http://en.wikipedia.org/wiki/Clinical_decision_support_system)
12. Cios KJ, Moore GW: Uniqueness of medical data mining. *Artif Intell Med* 2002; 26(1-2): 1-24.
13. Booch G, Maksimchuk RA, Engel MW, Young BJ, Conallen J, Houston KA: *Object-oriented Analysis and Design with Applications*, 3rd ed. Upper Saddle River 2007: Addison-Wesley.
14. Document-oriented database. In: *Wikipedia, The Free Encyclopedia*. San Francisco 2011: Wikimedia Foundation. [http://en.wikipedia.org/wiki/Document-oriented\\_database](http://en.wikipedia.org/wiki/Document-oriented_database)
15. 10gen, Inc.: *mongoDB*. <http://www.mongodb.org>
16. Object-relational database. In: *Wikipedia, The Free Encyclopedia*. San Francisco 2011: Wikimedia Foundation. [http://en.wikipedia.org/wiki/Object-relational\\_database](http://en.wikipedia.org/wiki/Object-relational_database)
17. Graph database. In: *Wikipedia, The Free Encyclopedia*. San Francisco 2011: Wikimedia

- Foundation.  
[http://en.wikipedia.org/wiki/Graph\\_database](http://en.wikipedia.org/wiki/Graph_database)
18. Neo4j.org: Neo4j: NOSQL for the Enterprise.  
<http://neo4j.org>
  19. EN 13606. In: *Wikipedia, The Free Encyclopedia*. San Francisco 2011: Wikimedia Foundation.  
[http://en.wikipedia.org/wiki/EN\\_13606](http://en.wikipedia.org/wiki/EN_13606)
  20. openEHR Foundation: *ISO EHR Standards*.  
<http://www.openehr.org/standards/iso.html>
  21. Garde S: *Archetypes: The Building blocks for Electronic Health Records*. Austin Health Seminar 2006: Central Queensland University.  
[http://esvc000269.wic061u.server-web.com/resources/informatics/austin\\_openEHR\\_Garde\\_compressed.pdf](http://esvc000269.wic061u.server-web.com/resources/informatics/austin_openEHR_Garde_compressed.pdf)
  22. openEHR Foundation: *Archetypes and Templates*.  
<http://www.openehr.org/releases/1.0.1/html/architecture/overview/Output/archotyping.html>
  23. Beale T: Archetypes: Constraint-based Domain Models for Future-proof Information Systems. In: Baclawski K, Kilov H (Eds.), *Eleventh OOPSLA Workshop on Behavioral Semantics: Serving the Customer*. Boston 2002: Northeastern University; 16-32.
  24. Parsing. In: *Wikipedia, The Free Encyclopedia*. San Francisco 2011: Wikimedia Foundation.  
<http://en.wikipedia.org/wiki/Parsing>
  25. Beale T, Heard S: *The openEHR Archetype System*. 2007: openEHR Foundation.  
[http://www.openehr.org/releases/1.0.2/architecture/am/archetype\\_system.pdf](http://www.openehr.org/releases/1.0.2/architecture/am/archetype_system.pdf)
  26. Ontology engineering. In: *Wikipedia, The Free Encyclopedia*. San Francisco 2011: Wikimedia Foundation.  
[http://en.wikipedia.org/wiki/Ontology\\_engineering](http://en.wikipedia.org/wiki/Ontology_engineering)
  27. International Health Terminology Standards Development Organisation (IHTSDO): *SNOMED CT (Systematized Nomenclature of Medicine-Clinical Terms)*. <http://www.ihtsdo.org/snomed-ct/>
  28. SOAP note. In: *Wikipedia, The Free Encyclopedia*. San Francisco 2011: Wikimedia Foundation.  
[http://en.wikipedia.org/wiki/SOAP\\_note](http://en.wikipedia.org/wiki/SOAP_note)
  29. Beale T: openEHR EHR Extract. 2011: openEHR Foundation.  
<http://www.openehr.org/wiki/display/spec/openEHR+EHR+Extract>
  30. Gillogley Services: *CEN 13606-1 Reference model*.  
[http://www.gillogley.com/ehr\\_cen\\_13606\\_reference\\_model.shtml](http://www.gillogley.com/ehr_cen_13606_reference_model.shtml)
  31. Ocean Informatics Wiki: *Archetype Editor*. 2007: Ocean Informatics.  
<http://wiki.oceaninformatics.com/confluence/display/TTL/Archetype+Editor>
  32. World Health Organization: *International Classification of Diseases (ICD)*. Geneva 2010: WHO. <http://www.who.int/classifications/icd/en/>
  33. World Health Organization: *International Classification of Primary Care, Second edition (ICPC-2)*. Geneva 2004: WHO.  
<http://www.who.int/classifications/icd/adaptations/icpc2/en/index.html>
  34. U.S. National Library of Medicine: *Unified Medical Language System (UMLS)*. Bethesda 2011: NLM.  
<http://www.nlm.nih.gov/research/umls/>
  35. Qamar R, Rector A: Most: A System To Semantically Map Clinical Model Data To SNOMED CT. In: *Proceedings of the First European Conference on SNOMED CT (SMCS'06)*. Copenhagen 2006: EU Network of Excellence Semantic Interoperability and Data Mining in Biomedicine; 38-43.  
<http://www.hiww.org/smcs2006/proceedings/7QamarSMCS2006final.pdf>
  36. Stanford Center for Biomedical Informatics Research: *The Protégé Ontology Editor and Knowledge Acquisition System*. Stanford 2011: Stanford University School of Medicine.  
<http://protege.stanford.edu/>
  37. openEHR Foundation: *openEHR Archetype Editor*. 2011: openEHR Foundation.  
[http://www.openehr.org/svn/knowledge\\_tools\\_dotnet/TRUNK/ArchetypeEditor/Help/index.html](http://www.openehr.org/svn/knowledge_tools_dotnet/TRUNK/ArchetypeEditor/Help/index.html)
  38. *Clinical Knowledge Manager*.  
<http://www.openehr.org/knowledge>
  39. *Open Source Initiative*. <http://www.opensource.org/>
  40. Kalra D, Forslund D: Open source health system. In: Demetriades J, Kolodner R, Christoperson G (Eds.), *Person-Centred Health Records: toward HealthePeople*. New York 2005: Springer; 169-185. USA <http://eprints.ucl.ac.uk/1591/1/A16.pdf>