

UNIVERZA V LJUBLJANI  
EKONOMSKA FAKULTETA

MAGISTRSKO DELO

**ONTOLOGIJA ŽIVLJENJSKIH SITUACIJ V  
OKVIRU E-UPRAVE**

Ljubljana, 23. januar 2009

DOMEN CUKJATI

### **IZJAVA**

Študent DOMEN CUKJATI izjavljam, da sem avtor tega magistrskega dela, ki sem ga napisal pod mentorstvom prof. dr. ANDREJA KOVAČIČA, in da v skladu s 1. odstavkom 21. člena Zakona o avtorskih in sorodnih pravicah dovolim njegovo objavo na fakultetnih spletnih straneh.

V Ljubljani, dne \_\_\_\_\_ Podpis: \_\_\_\_\_

## Kazalo vsebine

|                                                                   |           |
|-------------------------------------------------------------------|-----------|
| <b>UVOD.....</b>                                                  | <b>1</b>  |
| PODROČJE IN PROBLEM .....                                         | 1         |
| NAMEN .....                                                       | 2         |
| HIPOTEZE .....                                                    | 2         |
| CILJ.....                                                         | 2         |
| METODOLOGIJA DELA IN STRUKTURA POGLAVIJ .....                     | 2         |
| <b>1. E-UPRAVA IN MODELIRANJE ŽIVLJENJSKIH SITUACIJ .....</b>     | <b>4</b>  |
| 1.1. E-UPRAVA.....                                                | 4         |
| 1.1.1. Definicija e-uprave.....                                   | 4         |
| 1.1.2. Uporabniki e-uprave ter njihove težave.....                | 5         |
| 1.1.3. Semantična povezljivost e-storitev .....                   | 5         |
| 1.2. PORTALI ŽS.....                                              | 7         |
| 1.2.1. Koncept življenjske situacije.....                         | 7         |
| 1.2.2. »Front-office« in »back-office« e-uprave.....              | 8         |
| 1.2.3. Primerjava portalov ŽS .....                               | 9         |
| 1.2.4. Portal slovenske javne uprave: e-uprava.....               | 9         |
| 1.3. MODELIRANJE ŽIVLJENJSKIH SITUACIJ.....                       | 11        |
| 1.3.1. Tipologija modelov življenjskih situacij.....              | 11        |
| 1.3.2. Pregled pristopov k modeliranju ŽS .....                   | 12        |
| 1.3.3. Osnovni pojmi v modelu življenjskih situacij.....          | 15        |
| 1.3.4. Relacije v modelu življenjskih situacij .....              | 15        |
| 1.3.5. Zajem znanja o ŽS.....                                     | 16        |
| <b>2. PREDSTAVITEV ZNANJA.....</b>                                | <b>18</b> |
| 2.1. OSNOVNI POJMI .....                                          | 18        |
| 2.1.1. Znanje in upravljanje z znanjem .....                      | 18        |
| 2.1.2. Vrste znanja .....                                         | 18        |
| 2.1.3. Predstavitev znanja .....                                  | 20        |
| 2.1.4. Ontologija .....                                           | 20        |
| 2.1.5. Glavni vidiki pri PZ.....                                  | 20        |
| 2.1.6. Osnovne komponente .....                                   | 21        |
| 2.1.7. Tipi PZ .....                                              | 21        |
| 2.2. TEHNIKE PREDSTAVITVE ZNANJA .....                            | 22        |
| 2.2.1. Tehnike, ki ne podpirajo UI (1. skupina).....              | 23        |
| 2.2.2. Jeziki s področja UI (2. skupina).....                     | 24        |
| 2.2.3. Jeziki, temelječi na spletnih standardih (3. skupina)..... | 26        |
| 2.2.4. Povzetek tehnik za predstavitev znanja.....                | 28        |
| 2.3. OWL – WEB ONTOLOGY LANGUAGE .....                            | 29        |
| 2.3.1. Opis in razvoj .....                                       | 29        |
| 2.3.2. Različice jezika OWL ter razširitve .....                  | 30        |
| 2.3.3. Konstrukti jezika OWL.....                                 | 31        |
| 2.3.4. Konstrukt Class.....                                       | 32        |
| 2.3.5. Konstrukt Individual.....                                  | 34        |
| 2.3.6. Domneva odprtega sveta .....                               | 34        |
| 2.3.7. Delo z jezikom OWL .....                                   | 35        |
| 2.3.8. Akronim OWL.....                                           | 35        |
| 2.3.9. Potencial jezika OWL .....                                 | 35        |
| 2.4. UREJEVALNIKI ONTOLOGIJ .....                                 | 35        |
| 2.4.1. Splošno o urejevalnikih .....                              | 36        |
| 2.4.2. Orodja za OWL.....                                         | 36        |
| 2.4.3. PROTEGE-OWL – urejevalnik za OWL .....                     | 37        |
| 2.5. METODOLOGIJE ZA GRADNJO ONTOLOGIJ .....                      | 38        |
| 2.5.1. Primerjava metodologij za gradnjo ontologij .....          | 38        |

|                                                      |                                                                |           |
|------------------------------------------------------|----------------------------------------------------------------|-----------|
| 2.5.2.                                               | Metodologija METHONTOLOGY.....                                 | 39        |
| <b>3.</b>                                            | <b>GRADNJA ONTOLOGIJE ŽIVLJENJSKIH SITUACIJ .....</b>          | <b>44</b> |
| 3.1.                                                 | SPECIFIKACIJA PRIHODNJE ONTOLOGIJE .....                       | 44        |
| 3.1.1.                                               | Namen ontologije.....                                          | 44        |
| 3.1.2.                                               | Uporabniki in uporaba znanja.....                              | 44        |
| 3.2.                                                 | ANALIZA OKOLJA .....                                           | 46        |
| 3.3.                                                 | ANALIZA IZVEDLJIVOSTI.....                                     | 46        |
| 3.3.1.                                               | Operativna izvedljivost.....                                   | 47        |
| 3.3.2.                                               | Tehnična izvedljivost .....                                    | 47        |
| 3.3.3.                                               | Časovna izvedljivost .....                                     | 47        |
| 3.3.4.                                               | Ekonomska izvedljivost.....                                    | 48        |
| 3.4.                                                 | ZBIRANJE ZNANJA.....                                           | 51        |
| 3.4.1.                                               | Raziskovalni projekti.....                                     | 51        |
| 3.4.2.                                               | Delujoči upravni portali .....                                 | 53        |
| 3.4.3.                                               | Ostala znanstvena literatura.....                              | 54        |
| 3.4.4.                                               | Zakonodaja.....                                                | 54        |
| 3.4.5.                                               | Uradniki.....                                                  | 55        |
| 3.5.                                                 | KONCEPTUALIZACIJA .....                                        | 55        |
| 3.5.1.                                               | Prvi korak: Slovar izrazov.....                                | 55        |
| 3.5.2.                                               | Drugi korak: Taksonomija konceptov.....                        | 56        |
| 3.5.3.                                               | Tretji korak: Diagram ad-hoc binarnih relacij.....             | 59        |
| 3.5.4.                                               | Četrty korak: Slovar konceptov .....                           | 60        |
| 3.5.5.                                               | Peti korak: Opis ad-hoc binarnih relacij.....                  | 61        |
| 3.5.6.                                               | Šesti korak: Opis atributov primerkov .....                    | 64        |
| 3.5.7.                                               | Sedmi korak: Opis lastnosti razredov .....                     | 64        |
| 3.5.8.                                               | Osmi korak: Opis konstant .....                                | 65        |
| 3.5.9.                                               | Deveti korak: Opis formalnih aksiomov.....                     | 65        |
| 3.5.10.                                              | Deseti korak: Opis pravil.....                                 | 65        |
| 3.5.11.                                              | Enajsti korak: Opis primerkov.....                             | 66        |
| 3.6.                                                 | FORMALIZACIJA MODELA .....                                     | 66        |
| 3.6.1.                                               | Razredi in njihove lastnosti.....                              | 67        |
| 3.6.2.                                               | Relacije .....                                                 | 68        |
| 3.6.3.                                               | Procedura za reševanje življenjske situacije »PorocimSe« ..... | 69        |
| 3.7.                                                 | IMPLEMENTACIJA MODELA .....                                    | 71        |
| 3.8.                                                 | VREDNOTENJE ONTOLOGIJE.....                                    | 72        |
| 3.9.                                                 | VZDRŽEVANJE ONTOLOGIJE .....                                   | 72        |
| 3.10.                                                | UPORABA ONTOLOGIJE.....                                        | 72        |
| <b>SKLEP .....</b>                                   | <b>74</b>                                                      |           |
| POVZETEK REZULTATOV DELA .....                       | 74                                                             |           |
| OVREDNOTENJE REZULTATOV DELA GLEDE NA HIPOTEZE ..... | 74                                                             |           |
| OVREDNOTENJE PRISPEVKA K ZNANOSTI .....              | 75                                                             |           |
| SMERNICE ZA NADALJNJE DELO .....                     | 75                                                             |           |
| <b>LITERATURA IN VIRI:.....</b>                      | <b>77</b>                                                      |           |
| <b>PRILOGA.....</b>                                  | <b>1</b>                                                       |           |

## Kazalo slik

|                                                                                                                                                                      |    |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Slika 1: UVEDBA KONCEPTA ŽIVLJENSKE SITUACIJE V JAVNO UPRAVO OLAJŠA UPORABNIKU IDENTIFICIRATI STORITVE JAVNE UPRAVE, KI SO POTREBNE ZA REŠEVANJE NJEGOVE ŽS. ....    | 8  |
| Slika 2: ORGANIZIRANJE ŽS V SKUPINE NA PORTALU E-UPRAVA. ....                                                                                                        | 10 |
| Slika 3: TIPOLOGIJA MODELOV ŽIVLJENJSKIH SITUACIJ. ....                                                                                                              | 11 |
| Slika 4: NIVOJI ORGANIZACIJE POJMOV V ONTOLOGIJI ŽIVLJENJSKIH SITUACIJ. ....                                                                                         | 15 |
| Slika 5: TABELA ZA OPIS ŽIVLJENJSKIH SITUACIJ, STORITEV JAVNE UPRAVE TER DOKUMENTOV. ....                                                                            | 16 |
| Slika 6: ANALIZA NEKEGA HIPOTETIČNEGA ZNANJA. ....                                                                                                                   | 20 |
| Slika 7: TRI SKUPINE TEHNIK ZA PREDSTAVITEV ZNANJA Z NEKAJ PREDSTAVNIKI. ....                                                                                        | 22 |
| Slika 8: RAZVOJ TEHNIK, KI NE PODPIRAJO UI. ....                                                                                                                     | 23 |
| Slika 9: TEHNIKE PREDSTAVITVE ZNANJA S PODROČJA UMETNE INTELIGENCE. ....                                                                                             | 25 |
| Slika 10: NA SPLETNIH STANDARDIH TEMELJEČI JEZIKI. ....                                                                                                              | 27 |
| Slika 11: AKTIVNOSTI PRI RAZVOJU ONTOLOGIJE. ....                                                                                                                    | 40 |
| Slika 12: KORAKI ZNOTRAJ AKTIVNOSTI »KONCEPTUALIZACIJA«, KOT JIH PREDLAGA METODOLOGIJA METHONTOLOGY. ....                                                            | 42 |
| Slika 13: TOK ZNANJA MED ONTOLOGIJO ŽIVLJENJSKIH SITUACIJ TER RAZLIČNIMI UPORABNIKI. ....                                                                            | 44 |
| Slika 14: TOČKA PRELOMA. ....                                                                                                                                        | 50 |
| Slika 15: GRAFIČNI MODEL ŽS »POROČIM SE«. ....                                                                                                                       | 52 |
| Slika 16: GRAFIČNI MODEL NAVIDEZNE STORITVE »PRIDOBITEV DOKUMENTOV DRŽAVLJANA«, KATERE REZULTAT STA DVA DOKUMENTA. ....                                              | 52 |
| Slika 17: GRAFIČNI MODEL NAVIDEZNE STORITVE »PRIDOBITEV DOKUMENTOV TUJCA«, KATERE REZULTAT JE VEČ DOKUMENTOV. ....                                                   | 53 |
| Slika 18: OPIS ŽIVLJENSKE SITUACIJE »POROČIM SE«. ....                                                                                                               | 54 |
| Slika 19: TAKSONOMIJA KONCEPTA ŽIVLJENJSKA SITUACIJA. ....                                                                                                           | 57 |
| Slika 20: TAKSONOMIJA KONCEPTA STORITEV. ....                                                                                                                        | 58 |
| Slika 21: TAKSONOMIJA KONCEPTA DOKUMENT. ....                                                                                                                        | 58 |
| Slika 22: TAKSONOMIJA KONCEPTA UPORABNIK. ....                                                                                                                       | 58 |
| Slika 23: TAKSONOMIJA KONCEPTA PROCEDURA. ....                                                                                                                       | 58 |
| Slika 24: TAKSONOMIJA KONCEPTA ELEMENTPROCEDURE. ....                                                                                                                | 58 |
| Slika 25: RELACIJE MED RAZREDI NA NAJVIŠJEM NIVOJU. ....                                                                                                             | 59 |
| Slika 26: RELACIJE MED PODRAZREDI RAZREDA ELEMENTPROCEDURE. ....                                                                                                     | 60 |
| Slika 27: DIAGRAM RAZREDOV (UML) ZA PODROČJE ŽIVLJENJSKIH SITUACIJ: RAZREDI NA NAJVIŠJEM NIVOJU, LASTNOSTI RAZREDOV, RELACIJE MED RAZREDI IN LASTNOSTI RELACIJ. .... | 63 |
| Slika 28: DIAGRAM PRIMERKOV (UML) ZA ŽIVLJENJSKO SITUACIJO »POROČIM SE«. ....                                                                                        | 63 |
| Slika 29: RAZREDI IN NJIHOVE TAKSONOMIJE V UREJEVALNIKU ONTOLOGIJ PROTEGE. ....                                                                                      | 67 |
| Slika 30: PRIMEREK RAZREDA ŽIVLJENJSKA SITUACIJA »POROCIMSE« TER VREDNOSTI NJEGOVH ATRIBUTOV. ....                                                                   | 68 |
| Slika 31: POVEZAVE MED PRIMERKI (OZ. RAZREDI), NJIHOVA TAKSONOMIJA TER LASTNOSTI. ....                                                                               | 69 |
| Slika 32: PRIMEREK PROCEDURE »PROCEDURAPOROCIMSE« IN RELACIJE DO DRUGIH PRIMERKOV. ....                                                                              | 70 |
| Slika 33: VIZUALIZACIJA PROCEDURE »PROCEDURAPOROCIMSE« S POMOČJO VTIČNIKA TGVIZ ZA UREJEVALNIK ONTOLOGIJ PROTEGE-OWL. ....                                           | 71 |

## Kazalo tabel

|                                                                                                                   |    |
|-------------------------------------------------------------------------------------------------------------------|----|
| TABELA 1: PRIMERJAVA PRISTOPOV K MODELIRANJU ŽIVLJENJSKIH SITUACIJ. ....                                          | 14 |
| TABELA 2: VRSTE ZNANJA OZ. KLASIFIKACIJE S PRIMERI. ....                                                          | 19 |
| TABELA 3: LASTNOSTI TEHNIK IN JEZIKOV ZA PREDSTAVITEV ZNANJA, KI SO PREDSTAVLJENI V TEM DELU. ....                | 29 |
| TABELA 4: PREGLED LASTNOSTI METODOLOGIJ ZA IZGRADNJO ONTOLOGIJ. ....                                              | 39 |
| TABELA 5: OPERATIVNA IZVEDLJIVOST. ....                                                                           | 47 |
| TABELA 6: TEHNIČNA IZVEDLJIVOST. ....                                                                             | 47 |
| TABELA 7: ČASOVNA IZVEDLJIVOST. ....                                                                              | 47 |
| TABELA 8: STROŠKI RAZVOJA TER OBRATOVANJA. ....                                                                   | 48 |
| TABELA 9: POSREDNE IN NEPOSREDNE KORISTI ZA NAROČNIKA TER UPORABNIKE. ....                                        | 49 |
| TABELA 10: NEPOSREDNI STROŠKI IN KORISTI ZA NAROČNIKA PO LETIH. ....                                              | 50 |
| TABELA 11: SLOVAR IZRAZOV ZA POTREBE GRADNJE ONTOLOGIJE ŽIVLJENJSKIH SITUACIJ. ....                               | 56 |
| TABELA 12: SLOVAR KONCEPTOV. ....                                                                                 | 61 |
| TABELA 13: TABELA AD-HOC BINARNIH RELACIJ. ....                                                                   | 62 |
| TABELA 14: ATRIBUTI PRIMERKOV. ....                                                                               | 64 |
| TABELA 15: FORMALNI AKSIOMI V ONTOLOGIJI ŽIVLJENJSKIH SITUACIJ. ....                                              | 65 |
| TABELA 16: PRAVILO V ONTOLOGIJI ŽIVLJENJSKIH SITUACIJ. ....                                                       | 66 |
| TABELA 17: POKRIVANJE INFORMACIJSKIH POTREB RAZLIČNIH UPORABNIKOV S STRANI ONTOLOGIJE ŽIVLJENJSKIH SITUACIJ: .... | 73 |

## Kazalo prilog

|                                                      |   |
|------------------------------------------------------|---|
| PRILOGA A: SPISEK POGOSTO UPORABLJENIH KRATIC: ..... | 1 |
|------------------------------------------------------|---|

# UVOD

V tem poglavju so predstavljeni področje, na katerega se nanaša pričujoče magistrsko delo, problem, ki ga poskuša delo reševati, namen, hipoteze in cilj dela ter uporabljena metodologija in struktura poglavij.

## PODROČJE IN PROBLEM

Razvoj javne uprave je v zadnjih desetletjih močno pod vplivom novega razumevanja njenega poslanstva. Težišče njenega delovanja se usmerja od izvajanja oblastne funkcije k reševanju situacij, v katerih se znajdejo državljani in podjetja, ki jih z eno besedo imenujemo uporabniki. V ospredje se postavljajo njihove potrebe in koristi. Javna uprava se trudi vzpostavljati stik z uporabniki, se jim prilagajati ter jim pomagati v njihovih situacijah. Med glavne koncepte, katerih namen je približati javno upravo uporabnikom, spadajo koncept »Vse na enem mestu«, koncept »Življenjska situacija<sup>1</sup>« ter uporaba sodobnih informacijsko-komunikacijskih tehnologij.

Koncept »Vse na enem mestu« pomeni, da uporabnik ne hodi od okenca do okenca, ko rešuje svojo situacijo, temveč opravi vse potrebne korake na enem mestu. Uslužbenec javne uprave na drugi strani okenca naj bi bil zato sposoben opraviti vse delo, ki se je prej opravilo na različnih okencih (Kubicek in Hagen, 2000), (Seidle, 1995), (Fountain et al., 1994), (Bent et al., 1999). Razumljivo je, da se v tem primeru od uslužbencev zahteva veliko širše poznavanje področja, kar zahteva dodatna izobraževanja ali pa nove pripomočke v obliki hitro dostopne baze znanja.

Življenjska situacija (v nadaljevanju ŽS) je prav tako koncept, ki služi približevanju javne uprave uporabnikom. Sam pojem ŽS je metafora, ki opisuje specifično situacijo ali dogodek v življenju posameznika, katerega reševanje lahko zahteva uporabo storitev javne uprave. Uporaba koncepta ŽS je le eden od načinov za urejanje in združevanje informacij in storitev, ki jih ponuja javna uprava, vendar se zdi Evropski komisiji najprimernejši (EC, 2003), ker najbolje pomaga pri implementaciji koncepta »Vse na enem mestu«.

Med sodobnimi komunikacijskimi potmi izstopa internet, ker je omogočil popolnoma nove načine komunikacije ter poslovanja in ker je med različnimi novimi komunikacijskimi potmi najbolj sprejet s strani uporabnikov (Schellog in Mans, 2004). Na osnovi interneta so se v devetdesetih letih prejšnjega stoletja najprej pojavile spletne predstavitve posameznih institucij javne uprave. Pozneje so se pojavili še osrednji državni portali, ki služijo kot enotne vstopne točke do vseh informacij, povezanih z javno upravo.

Zadnji trend pri gradnji portalov javne uprave je aktivni spletni portal javne uprave, ki temelji na konceptih »Vse na enem mestu« in »ŽS« (Leben in Kunstelj, 2004), (Momotko et al., 2007), (Vintar in Leben, 2002). Izraz aktivni pomeni, da portal prek aktivnega dialoga z uporabnikom prilagodi reševanje ŽS uporabniku oz. njegovim specifičnim okoliščinam. Pomaga mu torej sprožiti vse tiste in samo tiste storitve znotraj ŽS, ki so v

---

<sup>1</sup> Življenjska situacija (ang. life event), krajše ŽS. V nadaljevanju članka bo uporabljena kratica ŽS. Pod pojmom ŽS razumemo tako življenjsko situacijo državljana kot poslovno situacijo poslovnega subjekta.

dani situaciji pomembne za uporabnika in so obenem v pristojnosti javne uprave. Pri tem je cilj, da bi bil portal zasnovan tako, da bi uporabnik rešil svojo ŽS na enem mestu, prek ene vstopne točke, čeprav bi različne storitve znotraj ene ŽS seveda reševali različni ponudniki storitev. Nujen pogoj za uspešno implementacijo takega portala pa je podroben model ŽS.

Pregled stanja na področju modeliranja ŽS pokaže, da obstaja mnogo pristopov k modeliranju ŽS. Različnost pristopov izhaja iz različnih namenov uporabe teh modelov. Ker trenutno ni nekega centralnega pregleda nad znanjem o ŽS, je za vsako novo ŽS treba ponovno analizirati relevantne postopke od začetka. To predstavlja nepotrebno podvajanje naporov in stroškov, modeli sami niso poenoteni, so nepopolni, ozko usmerjeni, mnogokrat neažurni, poleg tega pa ne podpirajo implementacije aktivnega portala javne uprave (Cukjati et al., 2008).

## **NAMEN**

Glavni namen dela je pokazati, da je mogoče izdelati model ŽS, ki bi na enem mestu vključeval vse relevantne informacije o postopkih v javni upravi, ŽS in dokumentih, ki v njih nastopajo. Model ŽS bi poenotil terminologijo, omogočil dostop do znanja na enem mestu, omogočil uporabo tega znanja različnim tipom uporabnikov, izboljšal ažurnost informacij, preprečeval podvajanje znanja, podprl delovanje aktivnega portala ŽS ter olajšal (pocenil) razvoj e-storitev v okviru e-uprave.

## **HIPOTEZE**

Preverili bomo tri hipoteze:

- Nabor konceptov, ki za različne uporabnike znanja o ŽS zadovoljivo popišejo ŽS ter postopke slovenske javne uprave, je omejen.
- Izdelan konceptualni model je mogoče zapisati kot ontologijo in implementirati v obliki, ki jo lahko obravnavajo obstoječa, na spletnih tehnologijah temelječa, računalniška orodja.
- Razvita ontologija omogoča aktivno vodenje uporabnika pri reševanju njegove ŽS z upoštevanjem specifičnih okoliščin, v katerih je uporabnik.

## **CILJ**

Cilj dela je razvoj prototipa izvedljivega modela ŽS v slovenski e-upravi. Razvili bomo formalni model predstavitve znanja, ki bo konceptualizacija znanja s področja ŽS ter postopkov slovenske javne uprave. Zapisan bo kot ontologija ter implementiran v obliki, ki jo lahko obravnavajo obstoječa računalniška orodja. Glavno vodilo pri tem bo, da je model in nabor potrebnih tehnologij čim enostavnejši in da temelji na sprejetih standardih.

## **METODOLOGIJA DELA IN STRUKTURA POGLAVIJ**

Delo bo sestavljeno iz treh korakov: (1) analize področja e-uprave in modeliranja ŽS, (2) analize področja predstavitve znanja in (3) gradnje ontologije ŽS. Ti trije koraki predstavljajo tudi tri glavna poglavja v magistrskem delu.



### **1. korak: Analiza področja e-uprave in modeliranja življenjskih situacij**

Metodologija dela za ta korak bo vključevala pregled literature s področja modeliranja ŽS in postopkov javne uprave ter analizo obstoječih portalov ŽS. Znanje s področja javne uprave bomo poleg tega pridobili s proučevanjem normativnih aktov, ki urejajo delovanje javne uprave in analizo intervjujev z zaposlenimi v javni upravi, ki poznajo potek in izvedbo posameznih postopkov. Vse te metode se bodo uporabile s ciljem ekstrakcije seznama potrebnih pojmov in relacij oz. njihove prisotnosti ali odsotnosti. V prvem koraku bomo naredili seznam in opis osnovnih pojmov ter relacij, potrebnih za opis ŽS ter postopkov, kar bo predstavljalo načrt ontologije ŽS. Za vsak osnovni pojem bomo določili njegovo definicijo, opis, seznam pomembnih lastnosti in odredili njihov podatkovni tip. Določili bomo tudi vse pomembne relacije med osnovnimi pojmi, kar vključuje njihovo definicijo in opis.

### **2. korak: Analiza področja predstavitve znanja**

Metodologija dela za ta korak bo vsebovala analizo literature s področja predstavitve znanja, ki vključuje različne logike, jezike, ontologije, njihove implementacije ter orodja za njihovo gradnjo. S primerjalno analizo bomo izbrali najprimernejšo tehniko, jezik za njen zapis ter orodje za implementacijo. Kriteriji pri izbiri tehnike in jezika bodo možnost zapisa ontologije v obliki, ki jo lahko obravnavajo obstoječa računalniška orodja, izraznost ontologije, k njej priključeni sklepalni mehanizmi, obstoj orodij za izgradnjo ontologije, ki podpirajo ta jezik, primernost jezika za izmenjavo ontologij med aplikacijami, enostavnost integracije ontologije v aplikacijo ter združljivost jezika z ostalimi jeziki, ki se uporabljajo za predstavitev znanja na spletu.

### **3. korak: Gradnja ontologije življenjskih situacij**

Metodologija dela v tretjem koraku je empirični preizkus, kjer bomo konceptualni model, ki bo rezultat prvega koraka, formalno zapisali z jezikom, ki ga bomo izbrali v drugem koraku. Na ta način bomo zgradili ontologijo, ki bo zapisana v obliki, ki jo lahko obravnavajo obstoječa računalniška orodja.

# 1. E-UPRAVA IN MODELIRANJE ŽIVLJENJSKIH SITUACIJ

V tem poglavju so opisani e-uprava, ki predstavlja odgovor javne uprave na razvoj novih tehnologij, državni portali e-uprave, ki temeljijo na konceptu ŽS, ter modeliranje ŽS, ki je eden od pogojev za razvoj ravnokar omenjenih portalov.

## 1.1. E-UPRAVA

V tem delu je najprej opredeljen sam pojem e-uprava, nato so opisani tipi uporabnikov in težave, s katerimi se le-ti srečujejo. Na koncu je opisana semantična povezljivost e-storitev, ki predstavlja enega od obetavnejših pristopov k reševanju prej omenjenih težav in na katerega se nanaša tudi pričujoče magistrsko delo.

### 1.1.1. Definicija e-uprave

Obstaja veliko definicij e-uprave<sup>2</sup>, vendar pa bolj ali manj vse govorijo o prilagajanju uprave tehnologijam informacijske dobe in interneta z namenom boljšega upravljanja<sup>3</sup>. Pri tem ene definicije bolj poudarjajo IKT (informacijsko-komunikacijske tehnologije; angl. ICT – information communication technologies), druge pa spremembe v mišljenju in odnosih v celotnem javnem sektorju. Glede na tip uporabnikov ločimo med naslednjimi modeli komunikacije: G2C (angl. Government-to-Citizen ali Government-to-Customer), G2B (angl. Government-to-Business) ter G2G (angl. Government-to-Government)<sup>4</sup>.

Dobro implementirana e-uprava naj bi naredila sodelovanje med državljani in upravo, poslovnimi subjekti in upravo ter sodelovanje znotraj same uprave enostavnejše, hitrejše in cenejše. Podobno menita Atkinson in Castro (2008), da so glavne prednosti, ki jih prinaša vzpostavitev e-uprave, povečanje učinkovitosti, transparentnosti in dostopnosti storitev javne uprave.

Razvitost elektronskih storitev v okviru e-uprave merimo z različnimi večstopenjskimi modeli (Brown, 2003) (Palvia in Sharma, 2007), pri čemer vsaka višja stopnja vsebuje vse možnosti, ki jih ponujajo nižje stopnje:

- *Informacije*: na spletnih straneh so objavljeni npr. naslovi ustanov, spiski potrebne dokumentacije ter čas uradnih ur za določeno poslovalnico.
- *Enosmerne komunikacije*: uporabnik lahko shranjuje različne obrazce, ki so objavljeni na spletu, vendar jih mora potem po navadni pošti poslati naslovniku.
- *Dvosmerne interakcije*: uporabnik lahko uporabi interaktivne spletne obrazce z možnostjo overjanja, ki jih odda po elektronski poti.
- *Transakcije*: popolne elektronske storitve, ki vključujejo tudi odločanje o zadevi, elektronsko vročanje dokumentov ter elektronsko plačilo same storitve.

---

<sup>2</sup> E-uprava (angl. electronic government, e-government, e-gov, digital government, online government).

<sup>3</sup> <http://www.e.govt.nz/archive/about-egovt/about-e-govt-archived.html>

<sup>4</sup> Včasih je bil v uporabi izraz G2A (angl. Government-to-Administration).

### **1.1.2. Uporabniki e-uprave ter njihove težave**

Uporabnike e-uprave delimo na naslednje skupine (Smartgov, 2004):

- *Administratorji*: uslužbenci javne uprave, uradniki, informatiki, ki vzdržujejo e-upravo (ter e-storitve in e-obrazce) na tehnološkem nivoju. Lahko so tudi zunanji razvijalci e-storitev.
- *Eksperti*: eksperti področja javne uprave, ki so odgovorni za vsebinski del e-uprave. Eksperti razumejo e-upravo kot orodje, ki jim olajša izvajanje same uprave, za katero so odgovorni.
- *Menedžerji*: menedžerji javne uprave, včasih tudi za to področje odgovorni politiki, vidijo e-upravo kot analitično orodje, ki jim zagotavlja statistike in informacije za sprejemanje strateških odločitev za delovanje javne uprave.
- *Končni uporabniki*: državljani, poslovni subjekti ter zaposleni v javni upravi, ki uporabljajo e-upravo pri reševanju situacij.

Uporabniki e-uprave se pri uporabi e-uprave srečujejo z naslednjimi težavami (Smartgov, 2004):

- Kompleksnost področja, še posebej, če se želi upoštevati specifične okoliščine, ki lahko vplivajo na potek reševanja zadev.
- Težave pri zagotavljanju povezljivosti med obstoječimi informacijskimi sistemi znotraj enot in med njimi. Posebno zahtevno je povezovanje sistemov z organizacijami zunaj javne uprave.
- Pomanjkanje sistemov za pomoč končnim uporabnikom. Sistemi so lahko v obliki pomoči prek spleta, zbirk specifičnih informacij za posamezne storitve, reference na zunanje vire informacij, primeri itd.
- Pomanjkanje mehanizmov, ki bi zbirali znanje, ki se generira v javni upravi, in bi ga pretvarjali iz implicitne v eksplicitno obliko.
- Pomanjkanje skladnih procesnih modelov, ki bi pospeševali uporabo e-storitev.
- Organizacijske ter kulturne ovire, kot so strah zaposlenih pred novimi tehnologijami in metodami.

V projektu SmartGov (2004) so za reševanje teh težav predlagali dva sklopa ukrepov:

- Integracija novonastalih standardov z najnovejšimi tehnologijami ter zadnjim razvojem na področjih, kot so upravljanje z znanjem, spletne tehnologije ter ukrepi za doseganje povezljivosti.
- Vpeljevanje teh tehnologij na sistematičen način prek novih in poenotениh procesnih modelov ter procesnega reinženiringa.

Tudi sam kot avtor pričujočega dela menim, da bi se večina težav vsaj delno omilila, če bi obstajal osrednji, poenoten in širše sprejet model ŽS, ki bi zajemal in na enem mestu ponujal informacije, ki so potrebne za reševanje ŽS ali za izvajanje e-storitev, ki rešujejo določeno ŽS. Tako predstavlja ontologija ŽS, ki jo predlagam v nadaljevanju, enega od neposrednih pripomočkov za nadaljnji razvoj e-uprave.

### **1.1.3. Semantična povezljivost e-storitev**

Povezljivost e-storitev predstavlja enega večjih problemov e-uprave. Kaj sploh je povezljivost? Različni viri, npr. ISO/IEC JTC1 SC36, Gartner, 2007, Bekkers, 2005 ali Peristeras et al., 2006, jo različno definirajo, v pričujočem delu pa bo uporabljena s strani

IDABC<sup>5</sup> predlagana definicija (IDABC, 2008), ki se v EU najpogosteje uporablja: »Povezljivost pomeni sposobnost IKT-sistemov kot tudi poslovnih procesov, ki jih ti sistemi podpirajo, da lahko izmenjujejo podatke, informacije ter znanje.«

Kot obstaja več definicij povezljivosti, obstaja tudi več delitev le-te. Ker smo uporabili že definicijo, ki jo uporablja IDABC, uporabimo še delitev, ki jo le-ta ponuja (IDABC, 2004):

- *Tehnična povezljivost* vključuje tehnična vprašanja pri povezovanju računalnikov ter računalniških sistemov, kot so odprti vmesniki, integracija in zapis podatkov, varnost ter dosegljivost.
- *Semantična povezljivost* vključuje znanje in njegov zapis v taki obliki, da je razumljivo računalniškemu agentu različnih aplikacij.
- *Organizacijska povezljivost* vključuje tista pravna, organizacijska in psihosociološka vprašanja, ki jih je treba rešiti, če želimo omogočiti povezovanje različnih ljudi, oddelkov, organizacij ali držav.

Bekkers (2005) identificira več problemov na področju povezljivosti, ki morajo biti rešeni, pri integraciji e-storitev javne uprave, pri čemer na nivoju semantične povezljivosti izpostavlja pomanjkanje oz. neobstoje ontologij storitev in splošnih podatkovnih definicij. Ker bodo informacije o e-storitvah vključene v ontologijo ŽS, se pričujoče delo nanaša predvsem na reševanje problemov nivoja semantične povezljivosti.

Državne uprave se običajno lotevajo vzpostavljanja oz. zagotavljanja povezljivosti prek nacionalnih interoperabilnostnih<sup>6</sup> okvirjev (angl. Government Interoperability Framework – GIF) ali nacionalnih arhitektur (National Enterprise Architecture – NEA) (United Nations, 2007).

V primeru interoperabilnostnih okvirjev (GIF) se poskuša doseči interoperabilnost prek sprejemanja in definiranja standardov ter navodil, kako naj organi javne uprave, državljani ter podjetja komunicirajo med seboj. GIF sam vsebuje osnovne tehnološke specifikacije, ki se jih morajo organi javne uprave nujno držati (Guijarro, 2007).

V primeru nacionalnih arhitektur (NEA) pa se poskuša doseči interoperabilnost preko strateškega načrtovanja enotne arhitekture, ki vzpostavlja povezave med vladnim IKT ter funkcijami, ki jih le-ta podpira. Povedati je tudi potrebno, da medtem, ko nekateri jasno ločijo med GIF-i in NEA-mi, se drugim zdita obe iniciativi tesno in neločljivo povezani. Tak primer je SAGA (Germany's Standards and Architecture of e-Government Applications) (SAGA, 2003), ki vsebuje arhitekturo ter standarde za interoperabilnost hkrati. Poleg tega pa nekateri avtorji s pojmom NEA opisujejo tako GIF kot NEA in ne ločijo med temi pojmi (Janssen in Hjort-Madsen, 2007). Tako zaenkrat ne obstaja širše sprejeta definicija in ločitev med tema dvema pojmom.

Ne glede na definicijo pojmov pa morajo biti GIF-i in/ali NEA-e nujno predlagani oz. vsiljeni s strani države, sicer zasebnih ponudnikov elektronskih storitev ne upoštevajo, ker imajo vsaj večji med njimi interes obdržati monopolne pozicije, ki jih sprejetje GIF-a ali

---

<sup>5</sup> IDABC – Interoperable Delivery of European eGovernment Services to public Administrations, Businesses and Citizens.

<sup>6</sup> Povezljivost in interoperabilnost uporabljamo kot sopomenki. V Sloveniji lahko zasledimo uporabo obeh izrazov.

NEA-e razbija. Slovenija je vzpostavila Nacionalni interoperabilnostni okvir (Šel in Gabrijel, 2007).

## 1.2. PORTALI ŽS

V tem poglavju so predstavljeni portali javne uprave, ki temeljijo na konceptu ŽS. Najprej je razložen sam koncept ŽS, nato umestitev portala ŽS v elektronski sistem uprave, primerjava obstoječih portalov ŽS ter opis slovenskega portala e-uprava.

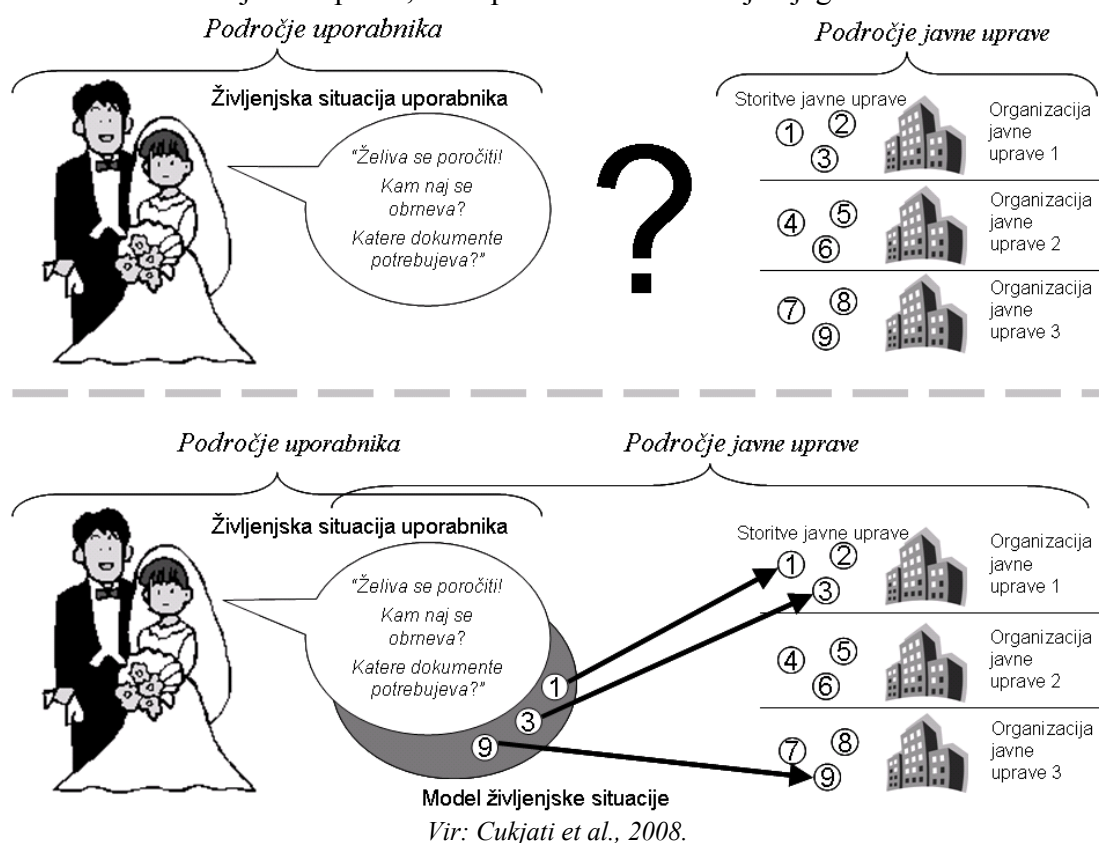
### 1.2.1. Koncept življenjske situacije

Življenjska situacija (v nadaljevanju ŽS) je metafora, ki opisuje specifično situacijo ali dogodek v življenju posameznika ali življenjskem ciklusu podjetja, katerih reševanje zahteva uporabo storitev javne uprave (Slika 1) (Wimmer in Tambouris, 2002) (Todorovski et al., 2006). Po drugi strani pa je uprava osredotočena na posamezne storitve, ki jih mora po zakonu nuditi uporabnikom. Med tema dvema pogledoma je razkorak, ki ga čuti uporabnik. Po eni strani želi uporabnik rešiti svojo ŽS, po drugi strani pa uporablja posamezne storitve javne uprave, ki so popolnoma nepovezane med seboj (Tambouris et al., 2006). Zaželeno je, da bi uporabniki na enostaven način izrazili svoje potrebe oz. namere, pri tem pa bi jim koncept ŽS pomagal identificirati vse tiste storitve javne uprave, ki bi jih bilo treba uporabiti glede na specifične okoliščine, v katerih je uporabnik. Storitve javne uprave tako niso več najvišja enota v ponudbi javne uprave, temveč le koraki pri reševanju ŽS uporabnikov. Koncept ŽS pomaga javni upravi, da se približa uporabnikom in da lažje razumeta drug drugega (Slika 1). Glede na to, katere storitve zajame pojem ŽS, se v literaturi pojavljata dva različna pogleda na ŽS, ožji in širši. ŽS v širšem pogledu zajema celotno situacijo uporabnika, kar pomeni, da so za reševanje njegove situacije potrebne storitve tako javnega kot privatnega sektorja. Primer je ŽS »Poročim se«, pri reševanju katere je treba uporabiti storitve javne uprave (prijava sklenitve zakonske zveze) kot tudi privatnega sektorja (rezervacija gostilne, najem poročne obleke ...). V ožjem pogledu pa ŽS zajema le storitve javne uprave. V pričujočem magistrskem delu bom uporabil slednjo možnost.

Primer ŽS je »Poročim se« (Todorovski et al., 2006), ki se zgodi dvema, ki se želita poročiti. Ta ŽS je v Sloveniji dokaj enostavna, ker vključuje le dve storitvi javne uprave, ki se morata izvesti, da se ŽS reši. Prva od njiju je »Prijava sklenitve zakonske zveze«, kjer se preveri upravičenost bodočih zakoncev do zakonske zveze. Druga od njiju pa je »Registracija zakonske zveze«, ki vključuje tudi samo ceremonijo sklepanja zakonske zveze ter podpis določenih dokumentov. Vendar je treba povedati, da postopek poteka tako le za običajna sklepanja zakonske zveze, medtem ko je v drugih primerih treba izvesti še druge postopke. Tako je npr. v primeru, ko je eden od bodočih zakoncev mladoleten, treba pridobiti dodaten dokument »Dovoljenje za sklenitev zakonske zveze mladoletnika«. Ta dokument pa se pridobi z izvedbo postopka javne uprave »Pridobitev dovoljenja za sklenitev zakonske zveze mladoletnika«. Različnih okoliščin uporabnika, ki povzročijo take posebne situacije, pa je lahko celo več in lahko nastopajo v poljubnih kombinacijah, npr. mladoletnik, bližnji sorodnik, tujec itd., kar lahko zelo poveča nabor različnih variant postopka oz. zelo zaplete reševanje določene ŽS.

Primeri ostalih ŽS so »Preselil sem se«, »Otrok se je rodil«, »Izgubil sem dokument«, »Želim voziti avtomobil«, »Upokojil sem se«, »Izgubil sem službo« itd.

Slika 1: Uvedba koncepta življenjske situacije v javno upravo olajša uporabniku identificirati storitve javne uprave, ki so potrebne za reševanje njegove ŽS.



### 1.2.2. »Front-office« in »back-office« e-uprave

Poslovanje javne uprave običajno razdelimo na poslovanje znotraj upravnih organov ter na poslovanje s strankami (angl. back-office in front-office). Zaradi tega se je tudi informatizacija poslovanja v preteklosti delila na dve področji oz. je šla po dveh poteh.

Po eni strani se je informatiziralo procese javne uprave, kar vključuje vzpostavitev elektronskih registrov podatkov in izgradnjo uporabniških vmesnikov, ki jih uporabljajo uradniki. Pomembno dejstvo pri tem je še, da so se postopki informatizirali neodvisno eden od drugega, da so to delali različni izvajalci in da si je vsak posebej ustvaril malenkost različno gledanje na vsebino informatizacije. Zato so posamezni sistemi med seboj pogosto nepovezani in nepovezljivi.

Po drugi strani pa so začeli nastajati državni portali, ki naj bi pomagali uporabnikom – državljanom in pravnim osebam – pri poslovanju z javno upravo. V začetnih fazah, kjer so večinoma tudi najrazvitejše države, tak portal ponuja večinoma le informacije o postopkih in storitvah, ne omogoča pa izpeljave celotnega postopka prek spleta, razen za nekaj posebej izbranih postopkov. Portal javne uprave, temelječ na konceptu ŽS, bi pomenil velik vsebinski korak naprej v smeri resničnega ponujanja vseh storitev na enem mestu ob upoštevanju uporabnikovih specifičnih okoliščin.

### **1.2.3. Primerjava portalov ŽS**

Upravni portali ŽS so se začeli razvijati na podlagi koncepta javne uprave »Vse na enem mestu« (one-stop-government) ob koncu devetdesetih let prejšnjega stoletja. Razvoj upravnih portalov je podrobneje predstavljen v študiji (Reinermann in Lucke, 2000), ki v glavnem obravnava osnovne koncepte upravnih portalov ter različne tehnične vidike njihovega razvoja. V zadnjih letih smo priča ad-hoc razvoju cele generacije portalov ŽS in platform za javno upravo »Vse na enem mestu«, kot tudi projektov s tega področja (npr. IST eGov<sup>7</sup>, IST Infocitizen<sup>8</sup>, IST TerreGov<sup>9</sup>, IST OntoGov<sup>10</sup>, IST SmartGov<sup>11</sup> idr.), kar kaže na izredno zanimivost tega področja. Izkušnje omenjenih projektov kažejo, da resnična javna uprava »Vse na enem mestu« zahteva mnogo več kot le vzpostavitev osrednjega državnega javnoupornega portala ter navidezno integracijo storitev v ŽS.

Kot ugotavlja primerjalna študija obstoječih portalov ŽS (Leben et al., 2004), implementacija koncepta ŽS le-te večinoma obravnava kot nek skupek storitev, ki so vsebinsko povezane z ŽS. Taki portali ponujajo storitve v okviru ŽS v obliki preprostega seznama, ki pa dejansko ne odraža dejanskih potreb uporabnika, saj le-ta ne ve, katere storitve mora dejansko uporabiti v svoji specifični situaciji. Ponudba upravnih storitev na tak način odraža pasivni pristop k oblikovanju storitev na osnovi ŽS. Tak koncept uporablja tudi obstoječi Slovenski državni portal e-uprava<sup>12</sup>.

V nasprotju s tem pa se v zadnjem času uveljavlja t. i. koncept aktivnega portala ŽS (Vintar in Leben, 2002) (Momotko et al., 2007), kjer je ŽS obravnavana kot proces, ki vsebuje odločitvene vidike. Uporabnik skozi aktivni dialog s sistemom definira situacijo, ki ustreza njegovemu problemu, sistem pa ga prek procesa sprejemanja odločitev vodi do seznama ustreznih storitev in potrebnih dokumentov. Uporabnik tako postane aktivni partner v celotnem procesu reševanja svojega problema. Ena od takih ŽS v Sloveniji, ki je tudi elektronsko podprta, je ustanovitev podjetja, katere reševanje poteka prek ene točke, kar omogoča sistem eVEM<sup>13</sup>. Problem dosedanjih implementacij je, da običajno uspejo zadostiti samo večjemu delu uporabnikov, kakšnih manj pogostih primerov pa ne pokrijejo, tako da mora marsikateri uporabnik nekje na sredini postopka le tega nadaljevati na običajen način.

### **1.2.4. Portal slovenske javne uprave: e-uprava**

Državni portal Republike Slovenije e-uprava, katerega skrbnik je Ministrstvo za javno upravo (MJU), je vstopna točka do različnih informacij o državni in javni upravi. Namen portala je poleg posredovanja informacij o upravi tudi približati upravne storitve uporabnikom prek spleta in tako poleg klasičnih komunikacijskih poti, po katerih lahko državljani in poslovni subjekti dostopajo do storitev javne uprave, ponuditi dodatno, elektronsko pot za opravljanje storitev.

<sup>7</sup> eGOV: An Integrating Platform for Realising Online One-Stop Government: <http://www.egov-project.org/default.htm>

<sup>8</sup> InfoCitizen: <http://www.infocitizen.org/>

<sup>9</sup> TerreGov: Impact of eGovernment on Territorial Government Service: [http://www.terregov.eupm.net/my\\_spip/index.php](http://www.terregov.eupm.net/my_spip/index.php)

<sup>10</sup> OntoGov: Ontology-enabled e-Gov Service Configuration: <http://www.ontogov.com/>

<sup>11</sup> SmartGov: A Governmental Knowledge-based Platform for Public Sector Online Services: <http://www.smartgov-project.org/>

<sup>12</sup> E-uprava – Življenjski dogodki: <http://e-uprava.gov.si/e-uprava/zivljenjskeSituacije.euprava>

<sup>13</sup> <http://evem.gov.si/sp/>

Glavna lastnost portala je usmerjenost k ciljnim skupinam uporabnikov. Rešitev je narejena v obliki podportalov za državljane, za pravne osebe ter podportala za zaposlene v javni upravi.

Informacije o postopkih za državljane so zbrane v sklopih, imenovanih življenjski dogodki. Storitve so razdeljene po področjih tako, da spremljajo vse pomembnejše dogodke v življenju posameznika od rojstva prek šolanja in zaposlitve do upokojitve (Slika 2). Opisi storitev so oblikovani tako, da uporabnika na razumljiv in pregleden način vodijo skozi postopek, nato pa je za nekatere storitve uporabniku na voljo tudi vloga oziroma obrazec za opisano storitev. Za nekatere storitve je podana tudi povezava na e-storitev, če seveda obstaja.

*Slika 2: Organiziranje ŽS v skupine na portalu e-uprava.*

|                                                                                     |                                                                                                                                                                                                                                                                                                                                                     |                                                                                     |                                                                                                                                                                                                                                                                            |
|-------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | <b>DRUŽINA IN OTROCI</b><br>Kje lahko sklenem zakonsko zvezo, kakšne so pravice zunajzakonskih partnerjev, kako pridobim rojstni list, kako postopam po rojstvu otroka v tujini, kako uveljavljam starševske pravice in pravice otrok, uredim varstvo, postanem rejnik,...                                                                          |    | <b>PROMET, VOZNIKI, VOZILA</b><br>Kdaj lahko opravim voziški izpit, kje lahko opravim tehnični pregled vozila, kako registriram vozilo, kako lahko podaljšam veljavnost prometnega dovoljenja, kje lahko kupim tablico ABC, kje lahko opravim izpit za voditelja čolna,... |
|    | <b>OSEBNA STANJA IN DOKUMENTI, PRIJAVA PREBIVALIŠČA</b><br>Kje pridobim osebno izkaznico, potni list, kako spremenim osebno ime, kako se prijavim na novem naslovu, na kakšen način pridobim delovno knjižico, kako in kje pridobim digitalno potrdilo za e-poslovanje, kako lahko pogledam v lastne osebne podatke, ki jih država hrani o meni,... |    | <b>OKOLJE IN PROSTOR</b><br>Kje pridobim gradbeno dovoljenje, kako prijavim stalno/začasno prebivališče, kako odjavim stalno prebivališče, kdaj sem upravičen do neprofitnega stanovanja,...                                                                               |
|    | <b>ŠOLA, IZOBRAŽEVANJE, MLADI</b><br>Kje poteka predšolsko izobraževanje, kako vpišem otroka v osnovno šolo, kje se vpišem v srednjo šolo, gimnazijo, kako opravi poklicna matura ali zaključni izpit, kakšni so pogoji za možnost financiranja študija, kako pridobim štipendijo,...                                                               |    | <b>KMETIJSTVO, GOZDARSTVO, PREHRANA</b><br>Kako pridobim status kmeta, kako lahko kupim/prodam kmetijsko zemljišče, kmetijo, gozd, kje pridobim dovoljenje za športni ribolov na morju,...                                                                                 |
|  | <b>DELO IN ZAPOSLOITVE</b><br>Kako sklenem delovno razmerje, na kakšen način dobim delovno knjižico, kje pridobim informacijo o prostih delovnih mestih, kako se lahko zaposlim v javni upravi, kakšne pravice imam kot brezposelna oseba,...                                                                                                       |  | <b>KULTURA IN ŠPORT</b><br>Kako lahko postanem samozaposlen v kulturi, samostojni novinar, na kakšen način lahko ustanovim društvo, kako prijavim javno prireditev/shod,...                                                                                                |
|  | <b>PODJETNIŠTVO</b><br>Kako postanem podjetnik, kako in kje lahko ustanovim s.p., katere dokumente potrebujem pri registraciji podjetja, na kakšen način ustanovim gospodarsko družbo, kako priglasim osebno dopolnilno delo,...                                                                                                                    |  | <b>VARNOST, REŠEVANJE, VOJSKA, OROŽJE</b><br>Kako postanem poklicni vojak, na kakšen način lahko naznanim kaznivo dejanje, kako pridobim orožni list,...                                                                                                                   |
|  | <b>OSEBNE FINANCE IN DAVKI</b><br>Kako in kdaj napovem napoved za odmero dohodnine, kako oddam zahtevek za zmanjšanje davčne osnove od dohodka, kako oddam napoved za odmero davka od premoženja, kako lahko pogledam v davčno evidenco, kako lahko investiram v državne vrednostne papirje,...                                                     |  | <b>DRŽAVA IN DRUŽBA</b><br>Kako pridobim potrdilo o nekaznovanosti, kje in kdaj lahko dobim brezplačno pravno pomoč, kako lahko izvem ali sem volilni upravičenec, na kakšen način lahko pridobim informacijo javnega značaja, kako lahko sodelujem pri odločanju,...      |
|  | <b>ZDRAVJE IN NEGA</b><br>Kako se vključim v obvezno, prostovoljno zdravstveno zavarovanje, na kakšen način in kdaj pridobim kartico zdravstvenega zavarovanja, kako in katere osebnostne zdravnike si lahko izberem,...                                                                                                                            |  | <b>SMRT IN ŽALOVANJE</b><br>Kdaj lahko dedujem, kje lahko pridobim dovoljenje za pokop izven pokopališča, kako pridobim izpisek iz matične knjige umrlih, kako lahko uveljavljam pogrebno, posmrtno,...                                                                    |
|  | <b>SOCIALNO VARSTVO, INVALIDI, VOJNE PRAVICE</b><br>Kako pridobim denarno socialno pomoč, kdaj sem upravičen do socialno-varstvenih storitev, na kakšen način lahko uveljavim pravico do tolmača slovenskega znakovnega jezika, kako uveljavim pravico žrtve vojnega nasilja,...                                                                    |  | <b>TUJINA, POTOVANJA, VIZUMI</b><br>S katerimi dokumenti lahko prehajam mejo, kako lahko pridobim vizum, kdo in na kakšen način lahko pridobi slovensko državljanstvo, status begunca,...                                                                                  |

*Vir: Portal e-uprava, 2008.*

Skrbnik portala je pri njegovem vzdrževanju in nadgrajevanju očitno dejaven, kar potrjujejo tudi nagrade, kot je nagrada za portal z najlažjo uporabo<sup>14</sup>.

Glavna pomanjkljivost portala oz. na njem ponujenih informacij je, da ne omogoča aktivnega vodenja uporabnika pri reševanju njegove situacije ob upoštevanju vseh njegovih specifičnih okoliščin. Portal zaenkrat ne kaže namena, da bi pokrival vse možne tipe uporabnikov, ampak le najpogostejše od njih.

<sup>14</sup>SI: eGovernment portal awarded the 'simplest website to use' award. Najdeno na spletu 12. 10. 2008: <http://www.epractice.eu/document/268>

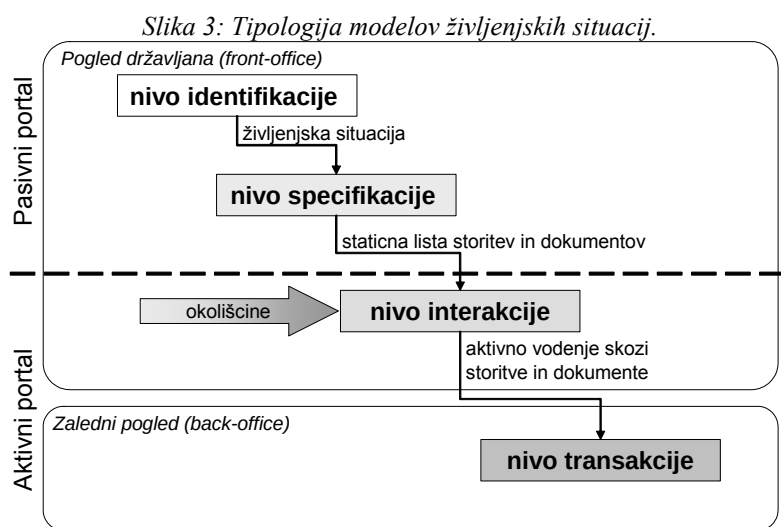


### 1.3. MODELIRANJE ŽIVLJENJSKIH SITUACIJ

Eden od nujnih pogojev pri gradnji portalov ŽS je dovolj dober model ŽS. V tem poglavju bo najprej predstavljena tipologija modelov ŽS, nato pregled pristopov k modeliranju ŽS, osnovni pojmi in relacije v modelu ŽS ter zajem znanja o ŽS.

#### 1.3.1. Tipologija modelov življenjskih situacij

Model ŽS je formalen opis same ŽS. Zaradi različnih motivov za gradnjo modelov ŽS, ki so bili predstavljeni v prejšnjem poglavju, se tudi modeli sami razlikujejo med seboj. Todorovski et al. (2006) ločijo štiri nivoje, na katerih je lahko posamezen model ŽS (Slika 3).



*Vir: Todorovski et al., 2006.*

Modeli ŽS na prvem nivoju (nivo identifikacije) vsebujejo le kratek opis ŽS. Ta opis omogoča uporabniku enostavno identifikacijo ŽS, v kateri je trenutno in ki jo želi rešiti. Opisi ŽS so običajno v obliki prostega teksta ter urejeni po področjih (npr. finance in davki, zdravje in nega) in/ali po ciljnih skupinah uporabnikov (npr. otroci, zaposleni, starejši ... majhna in srednja podjetja, velika podjetja, zavodi ...). Uporabniku sicer pomagajo pri identifikaciji ŽS, vendar vsebujejo premalo informacij, da bi mu lahko resneje pomagali tudi pri njihovem reševanju.

Modeli ŽS na drugem nivoju (nivo specifikacije) vsebujejo poleg informacij, ki jih vsebujejo modeli na identifikacijskem nivoju, še informacije o vseh storitvah javne uprave, ki so povezane z reševanjem določene ŽS. Informacije o storitvah običajno vsebujejo kratek in neformalen opis storitve, informacije o ponudniku storitev, spisek potrebnih dokumentov, tipičen čas reševanja zadeve, stroške za uporabnika ... Modeli na tem nivoju vsebujejo dovolj informacij za vzpostavitev pasivnega portala e-uprave. Vsebujejo namreč statično listo storitev, ki se nanašajo na določeno ŽS. Če obstajajo elektronske različice storitev, vsebujejo ti modeli tudi povezave nanje. Običajno vsebujejo še statično listo dokumentov, ki so potrebni pri sprožanju posameznih storitev javne uprave, in povezave nanje, če obstajajo njihove elektronske različice. Kljub vsem tem informacijam pa modeli ŽS na nivoju specifikacije ne morejo aktivno pomagati uporabniku pri reševanju njegove ŽS. Ti modeli namreč ne vsebujejo informacij o vseh uporabnikovih okoliščinah, ki bi lahko vplivale na reševanje ŽS. Zato ne morejo prilagajati seznama storitev in dokumentov konkretnim okoliščinam, v katerih je trenutno uporabnik.

Modeli ŽS na tretjem nivoju (nivo interakcije) odpravljajo omejitve modelov na prejšnjem nivoju in tako omogočajo razvoj aktivnih portalov e-uprave. Pogoji za to je, da se v model in opis posameznih storitev vključijo informacije o vseh možnih uporabnikovih okoliščinah, ki lahko kakorkoli vplivajo na izvajanje storitev in s tem posredno na reševanje ŽS. Uporabnikove okoliščine lahko vplivajo na reševanje ŽS na dva načina. Nabor storitev, ki so potrebne za reševanje določene ŽS, je odvisen od okoliščin, v katerih je trenutno uporabnik. V primeru ŽS »Poročim se« mora mladoletnik, ki se želi poročiti, uporabiti dodatno storitev, da pridobi dovoljenje za poroko od Centra za socialno delo. Po drugi strani pa okoliščine uporabnikov vplivajo na to, kateri dokumenti morajo biti priloženi vlogi za posamezno storitev. V primeru ŽS »Poročim se« mora tujec, ki se želi poročiti v Sloveniji, priskrbeti dodaten certifikat, s katerim dokazuje, da je tudi v svoji državi še vedno neporočen. Vse te informacije so sicer prisotne že v modelih na nivoju specifikacije prek povezav na zakonodajo, ki ureja področje posamezne ŽS. Prednost modelov na nivoju interakcije je, da so tukaj te informacije formalno in eksplicitno zapisane. Zato lahko tudi računalniški algoritem uporabi te informacije in z njihovo pomočjo prilagodi generično ŽS ter spisek potrebnih dokumentov specifičnim uporabnikovim okoliščinam.

Medtem ko se modeli ŽS na prvih treh nivojih posvečajo vidiku uporabnikov (front-office), modeli ŽS na četrtem (transakcijskem) nivoju upoštevajo še zaledni vidik storitev (back-office). Na tem nivoju vsebujejo modeli vse potrebne informacije, da se lahko storitve pri ponudnikih storitev tudi izvedejo. Ker posegajo na samo zaledno poslovanje znotraj ponudnikov javnih storitev, so zelo kompleksni, po drugi strani pa niso nujno potrebni za razvoj aktivnih portalov e-uprave. Zaradi tega se bomo v pričujočem magistrskem delu omejili na modele ŽS na interaktivnem nivoju.

### ***1.3.2. Pregled pristopov k modeliranju ŽS***

Problem modeliranja ŽS je izdelava formalnega opisa le-te, ki omogoča enostavno identifikacijo občana s konkretno ŽS, določa, katere storitve in procesi javne uprave so nujno potrebni za njeno razreševanje, ter določa, katere druge ŽS občana lahko vplivajo na njeno razreševanje. Gradnja formalnih modelov ŽS je zato zapleten proces, ki zahteva poznavanje vseh podrobnosti postopkovnega izvajanja vseh konkretnih storitev javne uprave, ki jih ŽS zajame.

Pregled novejšega stanja na področju modeliranja ŽS pokaže, da obstaja mnogo pristopov k problematiki modeliranja ŽS, ki pa izhajajo iz različnih potreb oz. je motivacija zanje zelo različna. Iz tega izhajajo razlike med njimi tako v širini, natančnosti kot vsebini modeliranja. Nekateri pristopi so usmerjeni na konceptualizacijo uprave v širšem smislu in se ŽS sploh ne dotaknejo ali pa le zelo površno kot Tomažič in Krisper (2001) ali Peristeras in Tarabanis (2005). Primerjava ostalih pristopov, ki se vsaj delno ukvarjajo s povezovanjem e-storitev v ŽS, je pokazala, da jih lahko delimo na dve veliki skupini:

V prvi skupini so raziskovalni pristopi, npr. projekti IST Infocitizen<sup>15</sup> (Adam et al., 2003), IST TerreGov<sup>16</sup> (Konstantas et al., 2006), IST OntoGov<sup>17</sup> (Apostolou et al., 2005), IST

---

<sup>15</sup> <http://www.infocitizen.org/>

<sup>16</sup> [http://www.terregov.eupm.net/my\\_spip/index.php](http://www.terregov.eupm.net/my_spip/index.php)

<sup>17</sup> <http://www.ontogov.com/>

SmartGov<sup>18</sup> (Georgiadis et al., 2002). Ti pristopi so zelo ambiciozni, saj ciljajo na povezovanje storitev na transakcijskem nivoju. Vendar se to pokaže kot slabost, ker dlje od demo verzij ne pridejo, kar je njihova glavna pomanjkljivost. Druga pomanjkljivost teh pristopov je, da do sedaj niso uporabljali pojma ŽS in posledično modela ŽS in zato manjkajo informacije o le-teh. To pomanjkljivost nameravajo nekateri preseči prek veriženja dokumentov, ki so potrebni v postopkih (Chun in Atluri, 2003). Vendar se v praksi ta pristop ne obnese, ker je semantika opisa storitev običajno preohlapna in ne zagotavlja potrebne gotovosti pri posredovanih informacijah. Nikoli namreč nismo povsem gotovi, da so v našem repozitoriju popisani vsi dokumenti, ki so potrebni za reševanje neke ŽS. Iz prakse pa vemo, da je to tako ali tako nemogoče. Projekt IST eGov<sup>19</sup> (Tambouris, 2001) je naredil še en korak v smeri ŽS, saj je vpeljal pojem ŽS, vendar zaradi ambicioznosti na drugih področjih ni razvil modela ŽS do te mere, da bi omogočal aktivni portal, oz. da bi upošteval specifične okoliščine, v katerih je lahko uporabnik. Podobno velja za pristop, ki ga predlagata Berčič in Vintar (2002), ki pa je ostal na nivoju predloga.

V drugo skupino spadajo pristopi, ki se uporabljajo v praksi. Ti pristopi imajo popolnoma drugačen vrstni red prioritet kot raziskovalni pristopi. V praksi je namreč najpomembnejše, da so rezultati takoj izvedljivi in uporabni. Obstajata dva načina modeliranja ŽS. Prvi je enostavno naštevanje postopkov, ki nastopajo v posamezni ŽS (Leben et al., 2004). Pri tem so ti spiski storitev ali preobširni, če so v spisku vključene vse storitve, ki bi lahko nastopale v posamezni ŽS, ali pa nepopolni, če se navajajo zgolj tiste storitve, ki pridejo v poštev za vse možne državljane, npr. v primeru ŽS »Poročim se« na portalu e-uprava<sup>20</sup>. Drugi način v praksi pa je, da se za vsako ŽS izdela svoja elektronska rešitev, ki pa ima znanje o ŽS posebej zapisano znotraj programa in ni dostopno ostalim uporabnikom, zaradi česar prihaja do večkratnega modeliranja postopkov in ŽS. Znanje je enkrat zapisano v aplikaciji (npr. e-VEM<sup>21</sup> za ustanovitev podjetja), drugič pa na portalu (npr. e-uprava<sup>22</sup>). V vseh primerih pa je bolj ali manj podrobno zapisano znanje o ŽS namenjeno le za določen način uporabe in ne omogoča ponovne uporabe zapisanega znanja.

Pristop, ki ga predlagajo Todorovski et al. (2006) odpravlja glavne pomanjkljivosti prejšnjih pristopov. To doseže tako, da zniža ambicije glede znanja o ŽS na interakcijski nivo, s čimer se zmanjša kompleksnost problema, še vedno pa model ŽS podpira aktivni portal ŽS. Večkratna uporabnost znanja je omogočena s primerno tehnologijo zapisa.

V spodnji tabeli (Tabela 1) je podana podrobnejša primerjava med pravkar omenjenimi pristopi k modeliranju ŽS, ki je povzeta po Cukjati et al. (2008). Kriteriji za primerjavo so nivo znanja, ki ga vsebuje model ŽS, ali pristop predvideva aktivni dialog in personalizacijo ŽS, ali implementacija omogoča večkratno uporabo zapisanega znanja, zrelost pristopa ter ostale prednosti in slabosti. Nivo znanja (nivo identifikacije, specifikacije, interakcije, transakcije), ki ga vsebuje model ŽS, je neposredno povezan z nivojem oz. lastnostmi končne rešitve. Od tega je odvisno, kaj bo končna rešitev nudila uporabnikom. Modeli ŽS na interakcijskem in transakcijskem nivoju lahko omogočajo aktivni dialog z uporabnikom in posledično personalizacijo ŽS ali storitev javne uprave, vendar ta možnost ni bila uporabljena v vseh pristopih. Zrelost pristopa (predlog, demo, v uporabi) opisuje stanje, v katerem je pristop. Dejstvo, da se pristop uporablja v praksi,

<sup>18</sup> <http://www.smartgov-project.org/>

<sup>19</sup> <http://www.egov-project.org/default.htm>

<sup>20</sup> <http://e-uprava.gov.si/e-uprava/dogodkiPrebivalci.euprava?zdid=75>

<sup>21</sup> <http://evem.gov.si/>

<sup>22</sup> <http://e-uprava.gov.si/e-uprava/dogodkiPrebivalci.euprava?zdid=1166&sid=802>

dokazuje njegovo uporabnost, po drugi strani pa lahko služi kot merilo za trenutno stanje v praksi. Večkratna uporaba zapisanega znanja je odvisna od načina implementacije modela ŽS in ni neposredno povezana z nobenim od prejšnjih kriterijev. Ta kriterij prikazuje pomembnost, ki so jo razvijalci pristopa namenili večkratni uporabi enega in istega znanja o ŽS. Kriterija prednosti in slabosti sta namenjena povzetku prednosti ali slabosti posameznega pristopa.

*Tabela 1: Primerjava pristopov k modeliranju življenjskih situacij.*

| Pristop                                                                                                        | Nivo znanja:<br>- identifikac.<br>- specifikacija<br>- interakcija<br>- transakcija | Aktivni dialog in personalizacija ŽS?                              | Implementacija omogoča večkratno uporabnost znanja? | Zrelost pristopa:<br>- predlog<br>- demo<br>- v uporabi | Glavne prednosti                                                                                              | Glavne slabosti                                                |
|----------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|--------------------------------------------------------------------|-----------------------------------------------------|---------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------|
| <b>Projekti IST:</b><br>- <i>EU-Publi.com</i><br>- <i>Infocitizen</i><br>- <i>OntoGov</i><br>- <i>SmartGov</i> | Transakcija (teoretično, ambicija)                                                  | -<br>(ŽS ni definirana, zato tudi ni znanja o ŽS)                  | ✓                                                   | Demo                                                    | Uporaba SWS, visoke ambicije (transakcijski nivo), XML model e-storitev                                       | Ne uporabljajo pojma življenjska situacija, samo demo          |
| <b>Veriženje prek dokumentov</b><br>( <i>Chun in Atluri, 2003</i> )                                            | Transakcija (teoretično, ambicija)                                                  | -<br>(ŽS ni definirana, zato tudi ni znanja o ŽS)                  | ✓                                                   | Demo                                                    | Uporaba ontologije področja                                                                                   | Ne uporablja življenjskih situacij                             |
| <b>IST eGov</b>                                                                                                | Transakcija (teoretično, ambicija)                                                  | -                                                                  | ✓                                                   | Demo                                                    | Definicija jezika GovML, uporaba pojma življenjska situacija                                                  | Ne vključuje uporabnikovih specifičnih okoliščin               |
| <b>Ontologija ŽS</b><br>( <i>Berčič in Vintar, 2002</i> )                                                      | Specifikacija                                                                       | -                                                                  | ✓                                                   | Predlog                                                 | Zapis v obliki ontologije                                                                                     | Pristop ostal na nivoju predloga                               |
| <b>Pristopi v praksi</b><br>- <i>e-uprava</i> <sup>23</sup>                                                    | Specifikacija                                                                       | -<br>Neke vrste personalizacija obstaja, vendar je skrita v tekstu | ✓                                                   | V uporabi                                               | Delujoč sistem                                                                                                | Zgolj tekstovni opis ŽS                                        |
| <b>Pristopi v praksi</b><br>- <i>eVEM</i> <sup>24</sup>                                                        | Transakcija                                                                         | ✓                                                                  | -                                                   | V uporabi                                               | Delujoč sistem                                                                                                | Znanje o ŽS je zakodirano v aplikacijo, zato nedostopno drugim |
| <b>Pristop</b><br><b>Todorovski et al. (2008)</b>                                                              | Interakcija                                                                         | ✓                                                                  | ✓                                                   | Predlog                                                 | Personalizacija, večkratna uporabnost modela, metodologija zbiranja znanja je bila uporabljena v projektu OSG | Šele predlog                                                   |

(Vir: Cukjati et al., 2008)

<sup>23</sup> <http://e-uprava.gov.si/e-uprava/dogodkiPrebivalci.euprava?zdid=1166&sid=802>

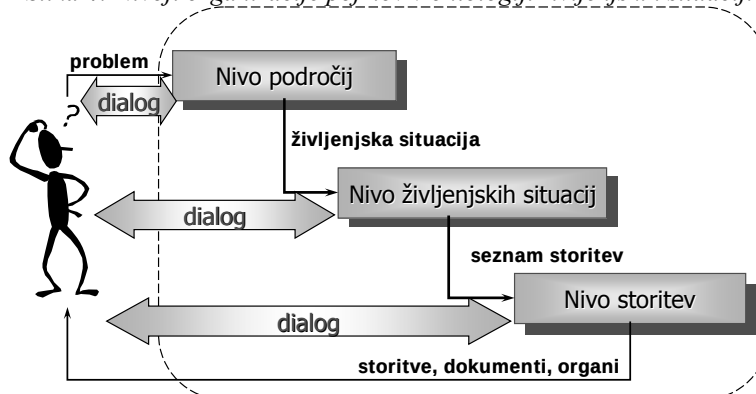
<sup>24</sup> <http://evem.gov.si/>

### 1.3.3. Osnovni pojmi v modelu življenjskih situacij

Obstoječe delo na tem področju predlaga organizacijo osnovnih pojmov na petih nivojih (Leben et al., 2004) (Slika 4):

- Področja in podpodročja javne uprave. Na tem nivoju so določena možna področja in podpodročja javne uprave, ki so osnovno vodilo pri identifikaciji ŽS. Primeri področij so recimo občani, družina in otroci, delo in zaposlitev ali kmetijstvo. V dosedanjih raziskavah so področja in podpodročja organizirana v hierarhijo, kar predstavlja resno omejitev glede na to, da so področja prepletena med sabo (in tako eno podpodročje lahko sodi v več področij hkrati) ter mnoge ŽS pripadajo več področjem hkrati. V primeru ontologije lahko podpodročja določa relacija »pripada področju« (glej razdelek o relacijah spodaj), kar omogoča mrežno organizacijo podpodročij in ŽS znotraj njih.
- Življenjske situacije. Na tem nivoju so zajete ŽS znotraj področij, določenih na prvem nivoju področij. Zajete so tako elementarne (npr. sprememba naslova stalnega bivališča) kot sestavljene (npr. selitev) ŽS. Odnos med njimi določa relacija »je sestavni del« (glej razdelek o relacijah spodaj). Ena od lastnosti ŽS je kompleksnost, saj je situacija lahko sestavljena ali elementarna.
- Storitve javne uprave. Na tem nivoju so zajete konkretne storitve javne uprave, kot je izdaja izvoda iz matične knjige rojenih. Zajete morajo biti tako storitve, ki se izvajajo tradicionalno, kot tudi e-storitve, tam kjer so te na voljo (kar je določeno z lastnostjo »tip dosegljivosti«).
- Organi javne uprave. V okviru tega tipa pojmov so v bistvu zajeti ponudniki storitev javne uprave. Z relacijami tipa »pripada področju« se lahko določi pripadnost organa določenemu področju, z relacijami tipa »ponuja storitev« pa se pove, katere storitve ponuja organ javne uprave.
- Dokumenti. Pomemben vidik modeliranja ŽS so dokumenti, ki so potrebni za razreševanje le-teh. Dokumenti se običajno pojavljajo kot predpogoj za izvedbo storitve (kar določa relacija tipa »potrebuje«) ali pa kot rezultati izvedbe storitve (kar določa relacija tipa »je rezultat«).

Slika 4: Nivoji organizacije pojmov v ontologiji življenjskih situacij.



Vir: Leben et al., 2004.

### 1.3.4. Relacije v modelu življenjskih situacij

Tipe relacij med pojmi, ki bi jih lahko zajela razvita ontologija, lahko razdelimo v naslednje skupine v odvisnosti od tipa povezave, ki je določena z relacijo:

- Organizacija javne uprave zajema relacije, ki so posledica organiziranosti javne uprave, neodvisno od ŽS. »X pripada področju P« označuje tip binarne relacije, ki določa domene posameznih področij, zajetih v ontologijo. Povezuje lahko podpodročje, upravni organ ali (sestavljeno) ŽS X s področjem Y (recimo podpodročje gradnja s področjem občani, če gre za gradnjo enodružinske hiše, ali s področjem podjetja, če gre za gradnjo poslovne stavbe) in s tem določi pripadnost pojma X področju Y. Binarna relacija tipa »UO izvaja S« določa, kateri upravni organ UO izvaja storitev S. Binarne relacije tipe »D je vhodni dokument S« (ali »D je izhodni dokument S«) povedo, kateri dokumenti so potrebni za izvajanje (oz. kateri dokumenti so rezultat izvajanja) storitve S.
- Odvisnosti med ŽS zajema relacije, ki povezujejo ŽS med sabo. Binarne relacije tipa »X je sestavni del Y« povezujejo sestavljeno ŽS Y z njenimi sestavnimi deli X, ki so zopet ŽS.
- Modeliranje ŽS zajema relacije, ki so pomembne za opis oz. modeliranje konkretnih ŽS. V tej skupini sodijo binarne relacije tipa »S je potrebna za razreševanje ZS«, ki določa nujnost opravljanja storitve S za razreševanje situacije ZS. Podobno relacija »S lahko razreši ZS« omogoča specifikacijo alternativnih storitev S, ki lahko razrešijo situacijo ŽS. Nenazadnje večmestne relacije tipa »S je potrebna za ZS pod pogojem P« bodo omogočale modeliranje kompleksnih pogojnih povezav med ŽS in storitvami, ker so potrebne storitve odvisne od drugih življenjskih situacij (recimo urejanje porodniškega dopusta pri rojstvu otroka je odvisno od zaposlitvenega statusa).

### 1.3.5. Zajem znanja o ŽS

Todorovski et al. (2006) predlagajo metodologijo za zajem znanja s področja ŽS, ki je potrebna za implementacijo aktivnega portala ŽS ob upoštevanju vseh možnih okoliščin državljana. Prvi korak pri gradnji modela, ki ga predlaga omenjena metodologija, je analiza ŽS, kjer se zbere vse potrebne informacije o ŽS, ki nas zanimajo. Pridobljene informacije se zapisujejo v tri vrste tabel (Slika 5). Tabele so prilagojene načinu razmišljanja uradnikov, ki poznajo področje in ki tabele tudi izpolnjujejo. Na ta način jim je olajšano izpolnjevanje tabel, je pa treba pozneje paziti, ker se nekateri podatki v tabelah podvajajo. Prva vrsta tabel se nanaša na opis ŽS, druga na opis posameznih storitev javne uprave (SJU), ki so potrebne pri reševanju ŽS, tretja pa na opis dokumentov. Za vsak dokument, ki je potreben ali pa nastopa kot rezultat postopka, se izpolni svoja tabela.

Slika 5: Tabele za opis življenjskih situacij, storitev javne uprave ter dokumentov.

| Tabela življenjskih situacij (ŽS) | Tabela storitev javne uprave (SJU)<br>(za vsako SJU znotraj posamezne življenjske situacije svoja tabela) | Tabela dokumentov<br>(za vsak dokument, ki nastopa v posamezni storitvi javne uprave svoja tabela) |
|-----------------------------------|-----------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|
| ID                                | ID                                                                                                        | ID                                                                                                 |
| Ime                               | Ime                                                                                                       | Ime                                                                                                |
| Kratek opis                       | Ponudnik                                                                                                  | Vloga                                                                                              |
| Povezana zakonodaja               | Kompetence ponudnika                                                                                      | Izdajatelj                                                                                         |
| Povezane SJU in ŽS                | Pravne podlage                                                                                            | Uporabnikove okoliščine                                                                            |
| Uporabnikove okoliščine           | Uporabnikove okoliščine                                                                                   | Interesent                                                                                         |
|                                   | Časovne odvisnosti                                                                                        |                                                                                                    |
|                                   | Prejemnik izhodov                                                                                         |                                                                                                    |
|                                   | Informacije o stroških                                                                                    |                                                                                                    |
|                                   | Čas izvajanja storitve                                                                                    |                                                                                                    |

V tabeli ŽS (Slika 5), so naslednji atributi: *ID* je edinstvena oznaka ŽS, *ime* je kratko ime ŽS, *kratek opis* pa je namenjen hitri predstavitvi ŽS, *povezana zakonodaja* našteva zakonodajo, ki določa reševanje ŽS, *povezane SJU in ŽS* pa so vse tiste SJU in ŽS, ki jih je morda treba izvesti pri reševanju določene ŽS. Katere pa so tiste okoliščine državljana, ki vplivajo na potek reševanja ŽS, je zapisano v zadnjem atributu *uporabnikove okoliščine*.

V tabeli SJU (Slika 5), ki opisuje neko storitev javne uprave, ki mora biti izvedena pri reševanju določene ŽS, so naslednji atributi: *ID* storitve, *ime* je kratko ime SJU, *ponudnik* pa je organ, ki storitev ponuja. *Kompetence ponudnika* so pomembne, če je več potencialnih ponudnikov, pravi pa je le eden, ki je v danem trenutku in okoliščinah pristojen za izvedbo storitve. *Pravne podlage* določajo zakonodajo, ki ureja posamezno storitev. *Uporabnikove okoliščine* pa podobno kot pri tabeli ŽS tudi tukaj opisujejo okoliščine, ki lahko spremenijo izvajanje storitve, le da je v tabeli SJU potrebno dodatno za vsako okoliščino navesti posledice te okoliščine s stališča uporabnika. *Časovne odvisnosti* so pomembne za gradnjo grafičnih modelov delovnih tokov, opisujejo pa situacije, ko mora biti katera storitev izvedena določen čas pred drugo. *Sprožitelj* je tisti, ki storitev sproži, *prejemnik izhodov* pa tisti, ki prejme izhode določene storitve, običajno dokumente. *Informacije o stroških* so koristne za uporabnika ter so običajno sestavljene iz taks in morebitnih dodatnih stroškov. *Čas izvajanja storitve* pa je maksimalni čas, v katerem mora biti storitev po zakonu opravljena. Lahko je dodan še povprečni čas, potreben za izvedbo storitve.

V tabeli dokumentov (Slika 5) je opisan dokument, ki je povezan s storitvijo, ki je bila opisana v prejšnji tabeli. Atributi so *ID*, ki je edinstvena oznaka dokumenta, *ime* je njegovo ime, *vloga* pa pove, kakšno vlogo ima ta dokument v sami storitvi. Možne izbire so *vloga*, če je to glavni vhodni dokument za storitev, *odločba*, če je to glavni izhodni dokument, ter *priloga*, če je to pomožni vhodni dokument, ki ga je treba priložiti h glavni vlogi. *Izdajatelj* pove, kdo dokument izda. *Uporabnikove okoliščine* pomenijo okoliščine, ob katerih je potrebno dokument priložiti glavni vlogi. Pomen imajo torej le za priloge. *Interesent* je uporabnik, ki potrebuje dokument.

## **2. PREDSTAVITEV ZNANJA**

V tem poglavju je predstavljeno področje predstavitve znanja. V pričujočem magistrskem delu želimo namreč razviti model ŽS ali povedano z drugimi besedami, želimo na trajen način predstaviti znanje o ŽS. Najprej bodo razdelani osnovni pojmi področja predstavitve znanja, nato bodo opisane in med seboj primerjane različne tehnike, ki se uporabljajo za predstavitev znanja, opisan bo jezik za predstavitev znanja, ki smo ga izbrali za svoje potrebe, opisani bodo urejevalniki ontologij, ki predstavljajo osnovno orodje, s katerim bomo delali, in na koncu bodo predstavljene še metodologije za gradnjo ontologij, od katerih si bomo izbrali najprimernejšo, po kateri bomo v nadaljevanju ontologijo ŽS tudi zgradili.

### **2.1. OSNOVNI POJMI**

V tem podpoglavju so predstavljeni osnovni pojmi področja predstavitve znanja. Tako so obrazloženi pojmi znanje, upravljanje z znanjem, vrste znanja, predstavitev znanja, ontologija, glavni vidiki pri predstavitvi znanja, osnovne komponente in tipi predstavitve znanja.

#### ***2.1.1. Znanje in upravljanje z znanjem***

Kaj je znanje? Preden odgovorimo na zastavljeno vprašanje, razčistimo dvojnost, ki se vleče že tisočletja. Največji filozofi so se in se še vedno ukvarjajo z vprašanjem, kaj je znanje, kar nakazuje, da področje ni povsem trivialno. Po drugi strani pa milijarde ljudi po vsem svetu vsakodnevno upravljajo z znanjem, ne da bi si enkrat samkrat v življenju zastavili vprašanje, kaj znanje sploh je. Imamo torej globoko filozofsko razumevanje tega pojma ter popolnoma praktično usmerjeno uporaba znanja. Kaj pa je vmes? Schwartz (2006) predlaga model, ki vključuje in pojasnjuje oba pogleda na znanje ter tudi ostale vidike, ki ju povezujejo ali ločujejo. Rečemo mu model upravljanja z znanjem.

Torej, kaj je znanje? Za naše potrebe je dovolj, če rečemo: »*Znanje je nekaj, kar vemo.*« Ena od drugih definicij pa definira znanje kot skupek urejenih informacij, ki privedejo do razumevanja.

#### ***2.1.2. Vrste znanja***

Ker obstaja veliko različnih pogledov na znanje, obstaja tudi več klasifikacij tega znanja, pri čemer se za različne dimenzije uporabljajo delitve z različnim številom kategorij (Alavi in Leidner, 2001) (Schwartz, 2006):



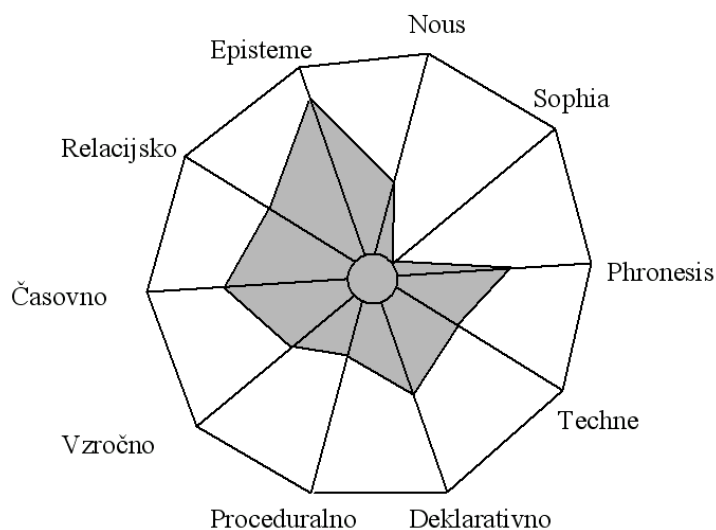
*Tabela 2: Vrste znanja oz. klasifikacije s primeri.*

| Dimenzija                  | Tip znanja                          | Definicija                                                                           | Primer                                                        |
|----------------------------|-------------------------------------|--------------------------------------------------------------------------------------|---------------------------------------------------------------|
| Izraženost                 | Tiho<br>(angl. tacit)               | Znanje, ki je osebno obravnavano in ga je težko formalizirati in posredovati naprej. | Osebna prepričanja o vzročno-posledičnih povezavah.           |
|                            | Izraženo<br>(angl. explicit)        | Znanje, ki je zapisano v formalnem jeziku in je zaradi tega prenosljivo.             | Znanje o pomembnih strankah v regiji.                         |
| Širina obstoja             | Splošno<br>(angl. common sense)     | Znanje, ki je prisotno v širši družbi.                                               | Upravna enota je del javne uprave.                            |
|                            | Strokovno<br>(angl. expert)         | Znanje, ki ga posedujejo le strokovnjaki področja.                                   | Kako pride do imenovanja načelnika upravne enote.             |
| Širina izvora              | Individualno<br>(angl. individual)  | Znanje, ki je nastalo in je neločljivo povezano z dejanji posameznika.               | Zaključki na podlagi izkušenj v različnih projektih.          |
|                            | Skupinsko<br>(angl. social)         | Znanje, nastalo in neločljivo povezano z dejanji skupine.                            | Norme in pravila skupinskega komuniciranja.                   |
| Kontekst                   | Deklarativno<br>(angl. declarative) | Znanje o dejstvih.<br>(angl. know-about)                                             | Katera vloga spada k določenemu postopku.                     |
|                            | Proceduralno<br>(angl. procedural)  | Znanje o postopanju.<br>(angl. know-how)                                             | Kako obravnavati prejeto vlogo.                               |
|                            | Vzročno<br>(angl. casual)           | Znanje o vzrokih.<br>(angl. know-why)                                                | Razumevanje, zakaj neka vloga ni bila obravnavana.            |
|                            | Časovno<br>(angl. conditional)      | Znanje o časovnih odvisnostih.<br>(angl. know-when)                                  | Znanje o tem, kdaj sprožiti naslednji korak postopka.         |
|                            | Relacijsko<br>(angl. relational)    | Znanje o povezavah med koncepti.<br>(angl. know-with)                                | Katere življenjske situacije so povezane z določenim stanjem. |
| Aristotelova klasifikacija | Episteme                            | Faktografsko ali znanstveno znanje. Skupaj z Nous tvorita znanje Sophia.             | Naslov določene upravne enote.                                |
|                            | Nous                                | Intuitivno znanje, ki ne uporablja pravil razumskega logičnega sklepanja.            | Določen uradnik mi je bolj naklonjen.                         |
|                            | Sophia                              | Teoretično znanje o univerzalni resnici in začetnem gibal.                           | Osnovni smisel mojega bivanja.                                |
|                            | Phronesis                           | Praktično znanje, temelječe na izkušnjah.                                            | Vrsta na upravni enoti je najkrajša v sredo dopoldan.         |
|                            | Techne                              | Veščine, temelječe na izkušnjah.                                                     | Pri pisanju odločbe začni stavek z: »Na podlagi ...«          |

*Vir: Delno povzeto po Alavi in Leidner, 2001.*

Na podlagi teh kriterijev, vseh ali samo nekaterih ali pa tudi kakšnih drugih, lahko za konkretno znanje naredimo preslikavo, ki nam poda grafično predstavitev narave nekega znanja. Za neko hipotetično znanje je na spodnji sliki (Slika 6) prikazana preslikava po zadnjih dveh dimenzijah, predstavljenih v prejšnji tabeli (Tabela 2).

Slika 6: Analiza nekega hipotetičnega znanja.



### 2.1.3. Predstavitev znanja

Schwartz (2006) v svojem modelu upravljanja z znanjem med drugim navaja pojem »Predstavitev znanja« kot eno od poglavij v celotnem področju upravljanja z znanjem. Mi bomo definirali predstavitev znanja kot: *»Predstavitev znanja je zapisovanje znanja na trajen način, ki omogoča kasnejši prenos znanja ter sklepanje človeku ali računalniku.«* Ta definicija razširja definicijo, ki se uporablja na področju umetne inteligence (Poole et al., 1998), s tem, da omili zahtevo po računalniškem sklepanju.

### 2.1.4. Ontologija

Cilj aktivnosti predstavitve znanja je na določen način predstavljeno znanje, čemur rečemo tudi ontologija. Čeprav obstaja mnogo različnih definicij pojma ontologija in bo minilo še nekaj časa, da se bodo pojmi ustalili in poenotili, naj na tem mestu podam verjetno najširše sprejeto definicijo pojma ontologija po Gruberju (1993): *»Ontologija je eksplicitna specifikacija konceptualizacije«*. Pozneje so definicijo razširili in podrobneje razložili (Studer et al., 1998): *»Ontologija je formalna, eksplicitna specifikacija dogovorjene konceptualizacije. Konceptualizacija se nanaša na abstraktni model nekega fenomena v svetu, kjer identificiramo relevantne koncepte tega fenomena. Eksplicitna pomeni, da so tipi uporabljenih konceptov in omejitve njihove uporabe eksplicitno definirani. Formalna se nanaša na dejstvo, da mora biti računalniško berljiva. Dogovorjena pa kaže na to, da ontologija vsebuje sporazumno znanje, torej znanje, ki je sprejeto od neke relevantne skupine.«*

### 2.1.5. Glavni vidiki pri PZ

Glavna vidika vsake tehnike predstavitve znanja sta po Corcho in Gomez (2000):

- *Izraznost*, ki pove, kaj lahko povemo o nečem neposredno iz statično zapisanega znanja. Pri tem je pomembno, kako oz. s katerimi komponentami je znanje zapisano.
- *Sklepalnost*, ki pove, do kakšnih novih zaključkov lahko pridemo s pomočjo sklepanja le na podlagi statično zapisanega znanja. Glavno vprašanje je torej, ali določena tehnika PZ podpira kakšnega od sklepalnih mehanizmov ali ne. Če mehanizem obstaja, potem se vprašamo še, ali zanj že obstaja avtomatičen sklepalnik. Ker živimo v dobi

računalnikov, nas zanimajo predvsem sklepalniki v obliki računalniških programov. Sami sklepalni mehanizmi so močno odvisni od uporabljenih komponent oz. načina, kako je znanje predstavljeno.

### 2.1.6. Osnovne komponente

Večino znanja se da zapisati s petimi formalno definiranimi komponentami, ki jih bolj ali manj uporabljajo skoraj vse tehnike za PZ, četudi zanje uporabljajo različna poimenovanja (Gruber, 1993) (Feier et al., 2005):

- *Koncepti* opisujejo osnovne pojme nekega področja. Definirani so prek njihovih hierarhij in atributov. Rang atributov so lahko podatki ali drugi koncepti.
- *Relacije* opisujejo odnose med različnimi parametri, kot so koncepti ali podatki. Tako kot koncepti imajo tudi relacije svoje attribute in včasih tudi hierarhije.
- *Funkcija* je poseben primer relacij, ki poleg tega, da povezuje attribute, vrne rezultat, ki je odvisen od atributov. Pri definiciji je treba torej določiti še aksiom, ki opredeljuje odvisnost rezultata od atributov.
- *Aksiomi* so logične izjave in pomagajo formalizirati znanje določenega področja.
- *Primerki* so utelešenje konceptov ali relacij. Definirani so lahko eksplicitno ali prek vrednosti atributov.

Nekatere tehnike, ki so sicer v manjšini, pa definirajo še procedure in/ali pravila:

- *Pravila* so izjave v obliki če – potem.
- *Procedure* so kompleksnejši zapisi med seboj povezanih pravil, ki opisujejo postopke.

Omeniti je treba, da imajo različni sistemi za PZ različna poimenovanja. Tako se npr. *relacije* v jeziku OWL imenujejo *properties*, v Protegu in okvirjih *slots*, v deskriptivnih logikah *roles*, v UML-ju ter ostalih objektno orientiranih sistemih *relations* in v GRAIL-u *attributes* (Horridge et al., 2004).

### 2.1.7. Tipi PZ

Različen nabor komponent za PZ, ki so v dani tehniki tudi formalno definirane, nam omogoča, da tvorimo različne tipe PZ:

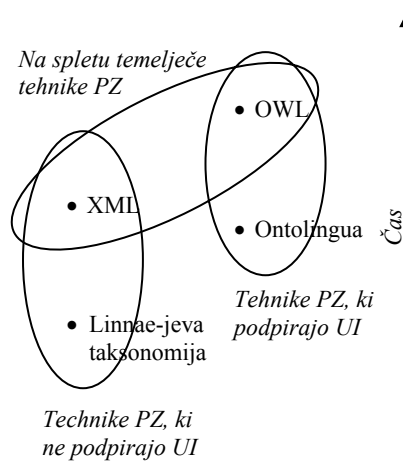
- *Neformalno besedilo* je tip PZ, kjer ni nobena komponenta formalno definirana. Zato je za kakršnokoli interpretacijo znanja potrebno dodatno znanje, ki je običajno shranjeno v glavi tistega, ki namera interpretirati besedilo. Iz istega razloga je kakršnokoli sklepanje možno le s strani ljudi, ker posedujejo dodatno znanje, ki opisuje zapisano znanje.
- *Slovar* je formalna lista izrazov z njihovimi definicijami. V splošnem vsebuje slovar razlago konceptov, ki so relevantni za določeno področje proučevanja ali uporabe. Če so tudi atributi formalno definirani, obstaja sklepalni mehanizem, ki mu pravimo avtomatična klasifikacija.
- *Taksonomija* predstavlja formalno definirane koncepte, ki so urejeni v hierarhično strukturo. Struktura sama po sebi je definirana prek formalno definiranih relacij nadrejeni-podrejeni, ki ji rečemo tudi relacija oče-sin. Če so formalno definirani še atributi konceptov, potem je možno izvajati sklepalna mehanizma kontrole dedovanja ter avtomatične klasifikacije.

- *Tezaver* razširja taksonomija tako, da uvede poleg hierarhičnih relacij še relacije ostalih vrst, kot so relacije enakovrednosti ali asociativnosti. Sklepalni mehanizmi pa so enaki kot pri taksonomiji.
- *Težkokategorna ontologija* vključuje poleg formalno definiranih konceptov in relacij še formalno definirane aksiome ali celo formalno definirane funkcije ter primerke. Ker na področju PZ prihaja do nekonsistentnosti glede izraza ontologija, saj nekateri s tem izrazom označujejo tudi tezaver in taksonomijo, bomo mi uporabljali definicijo, kjer uvrstimo tezaver in taksonomijo med lahkokategorne ontologije ter ostale ontologije med težkokategorne ontologije (Gomez et al., 2004). Težkokategorne ontologije tako omogočajo vse ostale poznane mehanizme sklepanja, kot so izvajanje procedur, kontrola omejitev in sklepanje na podlagi pravil.

## 2.2. TEHNIKE PREDSTAVITVE ZNANJA

Na različnih področjih uporabe se je razvilo veliko različnih tehnik za predstavitev znanja, ki jih lahko glede na različne vidike razvrstimo v različne skupine. Studer et al. (1998) tako predlagajo dve skupini tehnik, ki jih imenujejo lahkokategorne in težkokategorne ontologije in ki se razlikujejo po izraznosti tehnik. Lassila in McGuinness (2001) predlagata delitev po istem kriteriju, le da predlagata več nivojev. Gomez et al. (2004) so naredili podrobno primerjavo med različnimi jeziki za predstavitev znanja, vendar so se osredotočili le na jezike za zapis težkokategornih ontologij, ki so jih razdelili na dve skupini: tradicionalni jeziki in označevalni jeziki. Mi bomo delili tehnike v tri skupine, pri čemer so v prvi skupini tehnike, ki ne podpirajo umetne inteligence (UI)<sup>25</sup>, v drugi jeziki, ki se uporabljajo na področju UI, ter v tretji skupini jeziki, ki temeljijo na spletnih standardih. Takšna delitev ima zgodovinske osnove, saj poznamo tehnike prve skupine že od antičnih časov, tehnike druge skupine so se pojavile z iznajdbo računalnika pred pol stoletja, tretja skupina tehnik pa se je pojavila z uveljavitvijo interneta pred skoraj dvema desetletjema. Pomemben podatek je še, da sta prvi dve skupini disjunktni, tretja skupina pa vključuje tehnike iz obeh prvih dveh skupin (Slika 7).

Slika 7: Tri skupine tehnik za predstavitev znanja z nekaj predstavniki.

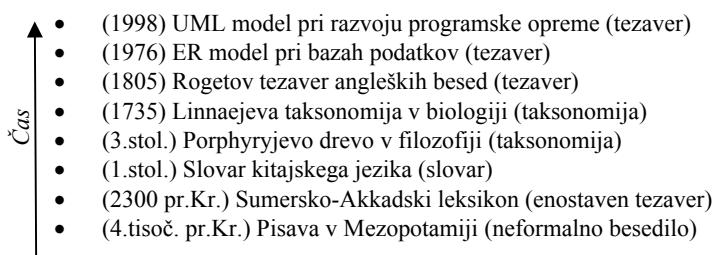


<sup>25</sup> UI – umetna inteligenca (angl. AI – artificial intelligence). Pomeni po eni strani vedo o inteligentnih strojih ter po drugi strani vedo o tehniki njihove izdelave.

### 2.2.1. Tehnike, ki ne podpirajo UI (1. skupina)

V tej skupini so vse tehnike PZ, katerih namen ni podpora umetni inteligenci, kar pomeni, da je v tej skupini večina tehnik, ki jih vsakodnevno uporabljamo. Glavna značilnost teh tehnik je, da se njihova izraznost prilagaja potrebam oz. da poskušajo pokriti potrebe predstavitve znanja v danem primeru. Ker naj bi jih uporabljali strokovnjaki posameznih področij in ne le strokovnjaki za predstavitev znanja, morajo biti čimbolj enostavne za uporabo. Z namenom enostavnosti in intuitivnosti uporabe so se znotraj mnogih tehnik razvili grafični vmesniki. Zaradi potrebe po enostavnosti te tehnike ne uvajajo veliko komponent, kar posledično pomeni malo možnih sklepalnih mehanizmov. Le avtomatična klasifikacija in preverjanje dedovanja sta lahko omogočena, pa še to samo v nekaterih tehnikah. Sklepanje s pomočjo teh tehnik je v veliki meri prepuščeno oz. omogočeno le človeškemu umu, ker le-ta poseduje dovolj dodatnih informacij, ki v samem zapisu znanja manjkajo. Te tehnike so se razvile že pred davnimi časi, vsekakor pa pred izumom računalnika. Po uveljavitvi spleta so hitro nastale različice teh tehnik, temelječe na standardih HTML ali XML. Tehnike 1. skupine omogočajo gradnjo neformalnih besedil, slovarjev, taksonomij ali tezavrov, vsekakor pa ne omogočajo gradnje težkokategornih ontologij.

Slika 8: Razvoj tehnik, ki ne podpirajo UI.



Pisava za tvorjenje neformalnih besedil je bila izumljena nekje v četrtem tisočletju pr. Kr. Nastala je zato, ker je obseg informacij pri trgovanju in državni administraciji začel presegati zmožnosti človeškega pomnjenja. Izum pisave je omogočil ogromen korak v smeri večje trajnosti in prenosljivosti znanja (Robinson, 2003). Formalno gledano je pisava reprezentacija jezika prek uporabe nekega nabora znakov. Običajno pomen besed v besedilu ni posebej oz. formalno definiran, temveč naj bi ga bralec že poznal. Zato lahko neformalno besedilo interpretirajo samo ljudje, ne pa tudi stroji ali računalniki. Zapis neformalnega besedila v računalniški obliki ne spremeni ničesar razen medija zapisa. Še vedno pa ostane vsa interpretacija besedila, npr. dela, ki ga beremo ta trenutek, na strani človeka.

Na področju lingvistike, ki je veda o jeziku, se je razvilo več tehnik za predstavitev znanja o besedah. Slovar (angl. vocabulary) je formalen spisek besed za neko potrebo. Ker so besede formalno definirane, jih lahko vidimo kot koncepte v naši terminologiji komponent. Slovar (angl. glossary) je nadgradnja prejšnjega z atributom »pomen besede«. Nadalje je slovar (angl. dictionary) nadgradnja prejšnjega še z dodatnimi atributi, kot so definicija, etimologija, izgovarjava besede ter ostale informacije o besedi. Prvi znani primer slovarja je *slovar kitajskega jezika* iz 1. stol. Vse vrste zgoraj omenjenih slovarjev s področja lingvistike (vocabulary, glossary, dictionary) spadajo med slovarje v naši klasifikaciji tipov PZ.

Slovar (angl. dictionary, lexicon) v lingvistiki ima lahko še en pomen, ki pomeni knjigo besed z ekvivalentnimi besedami v drugem jeziku. V tem primeru pa atribut, namesto da opisuje besedo samo, povezuje dve besedi, kar predstavlja povezavo ali relacijo »sinonim-v-jeziku-B«. Prvi tak znani slovar je *Sumersko-Akadski leksikon* iz okoli 2300 pr. Kr. Tak slovar je po naši klasifikaciji tezaver.

Tezaver v lingvistiki je različica slovarja (angl. lexicon), kjer so namesto atributa »sinonim-v-jeziku-B« dodani atributi »sinonim« in »antonim«, ki predstavljajo povezave na besede v istem jeziku. Prvi tezaver je bil *Rogetov tezaver angleškega jezika* iz leta 1805. Tezaver v lingvistiki je tezaver tudi v naši klasifikaciji.

Na področju filozofije je Porphyry v komentarju na Aristotelove kategorije že v 3. stol. naredil klasifikacijo stvari v obliki drevesa, zaradi česar je postala znana kot *Porphyryjevo drevo* (Sowa, 1987). Z drevesom je predstavljena metoda definiranja kategorij z določitvijo vrhnjega tipa ter diferenciatorjev, ki ločijo posamezne podtipе istega nadtipa. Porphyryjevo drevo je taksonomija v naši klasifikaciji.

V biologiji je bila leta 1735 predstavljena *Linneanova taksonomija* za klasifikacijo vseh živih stvari, ki so razvrščene v urejeno in formalno hierarhijo, ki se začne s kraljestvom. Drugi pomemben vidik sistema pa je predlagana splošna in sistematična binominalna nomenklatura za univerzalno poimenovanje organizmov. Linneanova taksonomija je taksonomija tudi v naši klasifikaciji.

Na računalniškem področju razvoja baz podatkov se je razvilo več tehnik, med katerimi je tudi *model ER* (model entiteta-povezava, angl. entity-relationship model) (Chen, 1976). Model je sestavljen iz grafičnega modela in podatkovnih slovarjev. Grafični model, imenovan tudi diagram, se uporablja za grafično predstavitev konceptov, imenovanih entitete, in relacij med njimi. Prednosti modela ER so formalna grafična reprezentacija, računalniška podpora ter enostavnost za uporabo. Model EER (Enhanced Entity-Relationship model) razširja osnovni ER z možnostjo določitve omejitev (Ramez in Navathe, 2007). Znanje, predstavljeno z modelom ER, je po naši klasifikaciji tezaver.

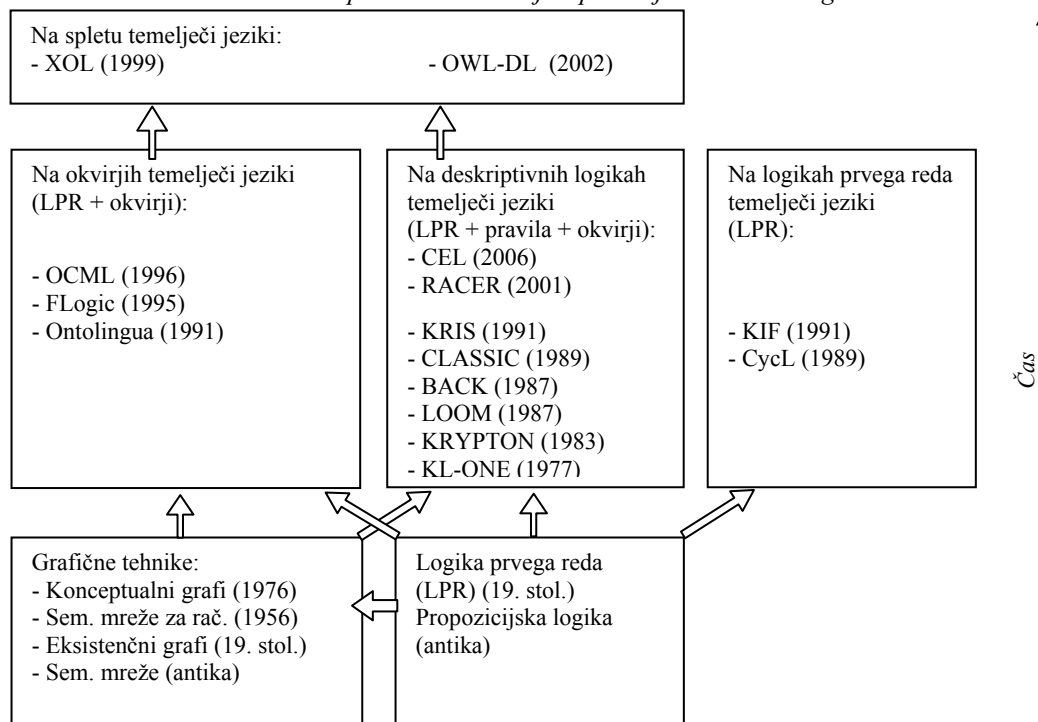
Na področju razvoja programske opreme je bilo tudi razvitih več tehnik, med katerimi je *UML* (Unified modelling language), ki predstavlja celo skupino diagramov (Rumbaugh et al., 1998). Med njimi sta tudi diagram razredov in diagram objektov, ki ju lahko uporabimo za predstavitev znanja, podobno kot model ER. Prednosti UML-ja so grafična predstavitev, računalniške osnove in množica računalniških orodij, ki jih podpirajo. Po naši klasifikaciji je UML tezaver. Da bi izkoristili prednosti UML-ja, obenem pa omogočili gradnjo tudi težkokategornih ontologij, je bil razvil OCL (Object constraint language) za definiranje aksiomov ali omogočanje dodatnih sklepalnih mehanizmov (Demuth in Hussmann, 1999). Vendar se je izkazalo, da je ugodneje nadgraditi že širše sprejete spletne jezike in jezike s področja umetne inteligence z grafičnimi vmesniki, kot pa grafičnim tehnikam UML-ja dodajati funkcionalnosti s področja umetne inteligence. Rezultat takega zlivanja funkcionalnosti je v nadaljevanju predstavljen OWL (Web Ontology Language).

### **2.2.2. Jeziki s področja UI (2. skupina)**

Umetna inteligenca (UI), prvič uporabljena leta 1956 (Crevier, 1993), pomeni tako vedo o inteligentnih strojih kot o tehniki njihove izdelave. Osrednji vidik UI je predstavitev znanja, saj je za vsak problem, ki naj bi ga reševali stroji, potrebno obširno znanje o

samem problemu. Torej je osnovna vloga PZ na področju UI predstaviti znanje na način, ki omogoča avtomatično sklepanje na podlagi tega znanja. Med stvari, ki naj bi bile predstavljene, spadajo objekti, njihove klasifikacije, lastnosti, relacije med njimi, okoliščine, stanja, dogodki, čas, vzroki in posledice itd. Ker je namen vseh tehnik PZ na področju UI uporaba znanja s strani računalnika, računalniki pa razumejo le jezike, pravimo tem tehnikam jeziki. Nekateri jeziki se osredotočajo bolj na logiko in sklepalne mehanizme, kot je na logiki prvega reda temelječ jezik KIF, drugi pa se osredotočajo na izraznost preddefiniranih konstruktov, kot je na deskriptivnih logikah temelječ jezik LOOM ali na okvirjih temelječ jezik Ontolingua (Slika 9). Vsi jeziki v tej skupini so po naši klasifikaciji namenjeni izgradnji težkokategornih ontologij.

Slika 9: Tehnike predstavitve znanja s področja umetne inteligence.



*Grafične tehnike*, kot so semantične mreže, eksistenčni grafi ali konceptualni grafi, lahko služijo tako za predstavitev znanja kot za podporo avtomatičnim sistemom pri sklepanju o zapisanem znanju. Sestavljene so iz okvirčkov, ki predstavljajo koncepte, ter puščic, ki predstavljajo semantične relacije med koncepti (Sowa, 1987). *Semantične mreže* so se že dolgo časa uporabljale v filozofiji, psihologiji, lingvistiki, svoje ponovno rojstvo pa so doživele na področju računalništva leta 1956 kot »interlingua«, vmesni jezik za prevajanje med naravnim govorom in računalniškimi jeziki. Logične opise lahko predstavimo tudi z *eksistenčnimi grafi*, predlaganimi konec 19. stol., ali *konceptualnimi grafi*, predlaganimi leta 1976 (Sowa, 1987). Slednji imajo izrazno moč enako ali večjo kot logika prvega reda. Grafične tehnike so bile zaradi razumljivosti vedno zanimive, zaradi česar mnogo grafičnih vmesnikov različnih jezikov temelji na predstavljenih tehnikah.

*Logike prvega reda (LPR)* so skupina logik, ki so formalno izpeljani sistemi in se uporabljajo v matematiki, filozofiji, lingvistiki in računalništvu. V nasprotju z angleščino LPR predstavljajo formalen in popolnoma nedvoumen jezik, zgrajen iz matematičnih struktur. LPR so sistem sklepanja, ki razširja propozicijske logike s tem, da dovolijo kvantifikacijo tudi prek individuumov. Propozicijske logike so bile predlagane že v

antičnih časih. *CycL*, prvič omenjen 1989, je na LPR temelječ jezik, s katerim je bila zgrajena baza znanja Cyc (Guha in Lenat, 1989). Besednjak CycL sestavljajo termini, ki jih lahko delimo na konstante, neatomarne termine, spremenljivke in še nekaj drugih tipov objektov. Termine nato povežemo v smiselne izjave CycL, s katerimi izpeljemo zaključke o znanju, zapisanem v bazi Cyc. Resničnost ali neresničnost stavkov CycL je odvisna od konteksta, kjer so konteksti predstavljeni v Cyc kot mikroteorije posameznih področij. *KIF* (Knowledge interchange format), predlagan 1991, je tudi na LPR temelječ računalniško usmerjen jezik za izmenjavo znanja med različnimi sistemi in programi (Genesereth, 1991). Ima deklarativno semantiko, kar pomeni, da je pomen izjav razumljiv tudi ljudem brez kakšnegakoli interpreterja. KIF omogoča predstavitev nemonotonih pravil za sklepanje ter definicijo objektov, funkcij in relacij.

*Okvirji*, predlagani leta 1974, so strukture, ki razdelijo znanje na podstrukture oz. manjše, zaključene enote. Nato združimo več okvirjev, da tvorimo celovite ideje. Vsak okvir vsebuje poleg znanja tudi informacije o tem, kako uporabiti okvir, kaj sledi in kaj storiti, če tisto, kar sledi, ne ustreza pričakovanjem. Okvirji so podobni objektom v objektno-orientiranem programiranju. Glavna prednost okvirjev v primerjavi z LPR je ta, da okvirji že vsebujejo preddefinirane koncepte za predstavitev znanja, kot je to v primeru jezika Ontolingua. *Ontolingua*, predlagana leta 1991, temelji na KIF-u in ontologiji okvirjev (Gruber, 1993), ki zagotavlja definicije konceptov, kot so razred, podrazred, primerek itd. *OCML* (Operational Conceptual Modelling Language), predlagan leta 1996, je operativna različica Ontolingue. *FLogic* (Frame Logic), omenjen leta 1995 (Kifer et al., 1995), je še en jezik s semantiko z močno teoretično osnovo ter utemeljeno in popolno teorijo dokazovanja.

*Deskriptivne logike* so družina jezikov za predstavitev znanja, ki so razširitev okvirjev in semantičnih mrež, ker slednje niso bile opremljene s formalno, na logiki osnovano, semantiko. Ime deskriptivne se po eni strani nanaša na deskripcijo konceptov za opis določenega področja in po drugi strani na semantiko, ki se jo da enostavno prepisati v LPR. Prvi sistemi deskriptivnih logik so bili *KL-ONE* (Brachman, 1977), *KRYPTON* (Brachman et al. 1983), *LOOM* (MacGregor in Bates, 1987), *BACK* (Luck et al., 1987), *CLASSIC* (Borgida et al., 1989), *KRIS* (Baader in Hollunder, 1991) ter pozneje *RACER* (Haarslev in Möller, 2001) in *CEL* (Baader et al., 2006). Danes so deskriptivne logike osnova semantičnega spleta pri izgradnji ontologij, saj na njih temelji eden od najbolj priljubljenih jezikov OWL-DL.

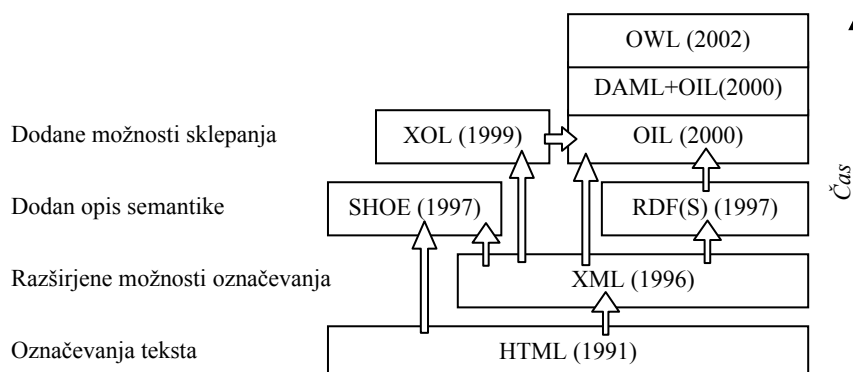
### **2.2.3. Jeziki, temelječi na spletnih standardih (3. skupina)**

Svetovni splet je kmalu po nastanku v začetku 90. let postal najbolj razširjen kanal za prenos podatkov in znanja. Zato so se kmalu razvili tudi jeziki za PZ, temelječi na spletnih standardih, kot je HTML. Po označevalnih jezikih za tekst jim pravimo tudi označevalni jeziki za ontologije. Njihova glavna prednost je, da jih lahko brez kakršnegakoli prevajanja postavimo na splet, pa so še vedno razumljivi avtomatičnim spletnim agentom. Ker pa je bil fokus pri njihovem razvoju usmerjen na spletne standarde, so bili drugi vidiki, kot sta izraznost in logika ali omogočanje sklepalnih mehanizmov, postavljeni v drugi plan. Tako sta HTML, ki ga tudi štejemo med jezike za PZ, ter XML namenjena le označevanju teksta oz. podatkov. RDF uvaja semantično opisovanje, medtem ko novejši jeziki, kot so XOL, OIL, DAML+OIL in OWL, nadgrajujejo prejšnje z značilnostmi jezikov s področja umetne inteligence (Slika 10). Zato včasih rečemo tudi, da so to jeziki UI, zapisani s sintakso XML ali RDF. RDF in SHOE omogočata izdelavo slovarjev, taksonomij in



tezavrov po naši klasifikaciji, medtem ko XOL, OIL, DAML+OIL in OWL omogočajo oblikovanje tudi težkokategornih ontologij.

Slika 10: Na spletnih standardih temelječi jeziki.



*HTML* (HyperText Markup Language), absolutno prevladujoč jezik za označevanje tekstov za spletne strani, je bil širše predlagan leta 1991 (Gillies in Cailliau, 2000). Predstavlja standard za označevanje teksta, zato tudi informacij v obliki teksta, ter omogoča opremljanje tega teksta z obliko, vstavljenimi slikami, povezavami in drugimi objekti. Njegova enostavna sintaksa in ogromen potencial, kar se tiče izmenjave informacij in dokumentov prek spleta, sta botrovala slavi ter sprejetju tega standarda. HTML je zaporedje ukazov, ki povejo, kako naj se prikaže predstavitevna stran. Ukazi se imenujejo tudi značke (tags). Značke so vedno med znakoma < in >, npr. <TAG>. Pri tem je nabor značk vnaprej nastavljen in ni razširljiv, kar je glavna omejitev HTML-ja kot jezika za PZ, ker ne omogoča označevanja teksta ali informacij po recimo semantiki, temveč le po nekaj oblikovnih lastnostih. Zaradi tega ne moremo izvajati nikakršne UI na tekstih, zapisanih v HTML-ju (Fensel, 2004).

*XML* (eXtensible Markup Language), prvič omenjen leta 1996, je splošni označevalni jezik, ki je bil razvit z namenom omogočanja izmenjave podatkov med različnimi informacijskimi sistemi. Temelji na HTML in SGML (Standard Generalized Markup Language), pri čemer je od prvega prevzel enostavnost, od drugega pa izraznost. Značke v XML niso več vnaprej definirane, temveč jih definira uporabnik (Fensel, 2004). XML tako omogoča izdelavo označevalnih jezikov za različne namene, npr. RSS (Really Simple Syndication), GovML (Government Markup Language), SHOE (Simple HTML Ontology Extension), RDF (Resource Description Language) itd. Sam po sebi XML ni namenjen PZ za potrebe UI, omogoča pa tvorjenje jezikov za ta namen.

*SHOE* (Simple HTML Ontology Extension), predlagan leta 1997, je v eni verziji razširitev jezika HTML, v drugi pa jezika XML. Narejen je bil z namenom semantičnega opisovanja informacij za objavo na spletu ter semantičnega iskanja po teh informacijah (Heflin in Hendler, 2000). Dodane so bile značke za opis dokumentov in informacij, ki jih spletni brskalniki ne prikazujejo. Te značke skupaj s SHOE domenskimi ontologijami bi bile lahko osnova za semantični splet, saj SHOE omogoča zapis entitet, atributov in relacij, ne omogoča pa zapisa kardinalnosti. Omogoča zapis taksonomij razredov entitet, ne omogoča pa dedovanja lastnosti na podrazrede. Velika slabost je tudi, da ne omogoča naprednejših sklepalnih mehanizmov, ker ne omogoča zapisa aksiomov oz. logike. Danes je SHOE bolj ali manj opuščena pobuda, ker so se pojavili drugi jeziki, ki pa velikokrat temeljijo na SHOE, npr. OWL, ki bo predstavljen v nadaljevanju.

*RDF* (Resource Description Framework), prvič omenjen leta 1997, je podobno kot *SHOE* en nivo nad *XML*. Originalno je bil razvit kot metapodatkovni model, vendar se je hitro uveljavil kot splošna metoda za modeliranje informacij. Metapodatkovni model *RDF* predlaga zapis o spletnih virih v obliki izjav subjekt-predikat-objekt, kot je v navadi v predikatni logiki. Tem izjavam rečemo v terminologiji *RDF* trojčki. Subjekt označuje vire, predikat označuje vidike virov ter izraža povezavo med subjektom in objektom. *RDF* sam po sebi je že dovolj za predstavitev izjav o določenem področju. Da lahko definiramo razrede, lastnosti, področja in obsege, pa si pomagamo s shemo *RDF* (*RDFS*) (Brickley in Guha, 1999), ki je na okvirjih temelječa razširitev *RDF*-a. *RDF* in *RDFS* skupaj poimenujemo *RDF(S)*. *RDF* je danes široko sprejet standard, vendar ima za potrebe PZ nekaj večjih pomanjkljivosti, kot so šibka izraznost trojčkov in dvoumnost pri izražanju trditev, kar je bilo pozneje rešeno z jezikom *OWL*.

*XOL* (*XML*-based *Ontology Exchange Language*), predlagan leta 1999 za potrebe izmenjave znanja med bioinformacijskimi ontologijami, je na okvirjih temelječ jezik s semantiko, podedovano od *OKBC-lite*, kjer je bila *Lisp*ova sintaksa nadomeščena s sintakso *XML*, da se je omogočila neposredna predstavitev na spletu. Pomanjkljivost *XOL*-a je, da omogoča zapis le razredov, njihovih hierarhij, atributov in primerkov (Corcho in Gomez, 2000). Kakorkoli, *XOL* je bil korak v pravo smer ter dobra motivacija za raziskovalce, da so se lotili prepisovanja s spletnimi sintaksami tudi drugih jezikov s področja *UI*. Tako so nastali *OIL*, *DAML+OIL* in *OWL*.

*OWL* (*Web Ontology Language*), *DAML+OIL* (*DARPA Agent Markup Language + OIL*) in *OIL* (*Ontology Inference Layer*) predstavljajo najmlajšo skupino jezikov, ki imajo namen omogočiti sklepalne zmožnosti *UI* in biti obenem zapisani s spletnimi sintaksami, kot sta *XML* ali *RDF*. Pri jeziku *OIL*, prvič omenjenem leta 2000, so semantika in sklepalni mehanizmi povzeti po deskriptivnih logikah in okvirjih (Fensel et al., 2000). Obstajata dve različici, ena s sintakso *XML* in druga s sintakso *RDF*. *DAML+OIL*, prav tako predlagan leta 2000, je naslednik *OIL* in je podprt s strani *W3C*. Vendar je *W3C* kmalu začel z lastno pobudo za razvoj jezika za zapis ontologij in tako je nastal *OWL*, ki je naslednik *DAML+OIL*. Ker je bil jezik *OWL* izbran kot najprimernejši jezik za potrebe pričujočega dela, je več o *OWL* zapisanega v posebnem poglavju, ki sledi temu poglavju. Tukaj omenimo le, da obstajajo tri verzije jezika, kjer je bila vsaka od njih razvita za svoj namen. *OWL Lite* je namenjen tistim, ki potrebujejo le funkcionalnosti za zapis klasifikacij in hierarhij ter enostavnih omejitev, *OWL DL* je bil narejen z namenom, da priskrbi maksimalno izraznost, vendar ohrani računsko popolnost, rešljivost ter razpoložljivost uporabnih sklepalnih mehanizmov, in zadnji, *OWL Full*, je bil razvit z namenom, da bi ohranil kompatibilnost z *RDF* oz. omogočil uporabo vseh konstruktov *OWL*, zaradi česar niti v teoriji ne obstaja računalniško orodje, ki bi bilo sposobno celotnega sklepanja nad znanjem, zapisanim z *OWL Full*.

#### **2.2.4. Povzetek tehnik za predstavitev znanja**

Tabela 3 prikazuje pregled značilnosti oz. lastnosti v prejšnjem poglavju predstavljenih tehnik in jezikov. V tabeli je z »●« označena lastnost, ki jo jezik ali tehnika podpira, z »-« lastnost, ki jo jezik ali tehnika ne podpira, z »D« da lastnost ni podprta, jo pa lahko dosežemo z nekaj dodatnega dela, in z »?« informacija, da podatek o lastnosti ni bil najden.

Tabela 3: Lastnosti tehnik in jezikov za predstavitev znanja, ki so predstavljeni v tem delu.

| Tehnika ali jezik za predstavitev znanja | Pisava v Mezopotamiji   | Sumersko-akkad. leksikon | Kitajski slovar       | Porphyryjevo drevo    | Linnaean-ova taksonomija | Tezaver angleških besed | ER model | UML model | Konceptualni grafi | LOOM (1987) | KIF  | Ontolingua | FLogic | OCML | HTML | XML  | SHOE | RDF(S) | XOL  | OIL  | OWL  |
|------------------------------------------|-------------------------|--------------------------|-----------------------|-----------------------|--------------------------|-------------------------|----------|-----------|--------------------|-------------|------|------------|--------|------|------|------|------|--------|------|------|------|
| Prva omemba ali pojavitev                | 4 <sup>th</sup> mil. BC | 3200 BC                  | 1 <sup>st</sup> cent. | 3 <sup>rd</sup> cent. | 1735                     | 1805                    | 1976     | 1998      | 1976               | 1987        | 1991 | 1991       | 1995   | 1996 | 1991 | 1996 | 1997 | 1997   | 1999 | 2000 | 2002 |
| Skupina                                  |                         |                          |                       |                       |                          |                         |          |           |                    |             |      |            |        |      |      |      |      |        |      |      |      |
| 1. skupina                               | •                       | •                        | •                     | •                     | •                        | •                       | •        | •         | -                  | -           | -    | -          | -      | -    | -    | -    | -    | -      | -    | -    | -    |
| 2. skupina                               | -                       | -                        | -                     | -                     | -                        | -                       | -        | -         | •                  | •           | •    | •          | •      | •    | -    | -    | -    | -      | -    | •    | •    |
| 3. skupina                               | -                       | -                        | -                     | -                     | -                        | -                       | -        | -         | -                  | -           | -    | -          | -      | -    | •    | •    | •    | •      | •    | •    | •    |
| Izraznost                                |                         |                          |                       |                       |                          |                         |          |           |                    |             |      |            |        |      |      |      |      |        |      |      |      |
| Koncepti                                 | -                       | •                        | •                     | •                     | •                        | •                       | •        | •         | •                  | •           | •    | •          | •      | •    | ?    | •    | •    | •      | •    | •    | •    |
| Primerki                                 | -                       | -                        | -                     | -                     | -                        | -                       | -        | -         | ?                  | •           | •    | •          | •      | •    | -    | ?    | ?    | •      | •    | •    | •    |
| Rel. (hierarh.)                          | -                       | -                        | -                     | •                     | •                        | -                       | •        | •         | •                  | •           | •    | •          | •      | •    | -    | -    | •    | •      | •    | •    | •    |
| Rel. (ostale)                            | -                       | •                        | -                     | -                     | -                        | •                       | •        | •         | •                  | •           | •    | •          | •      | •    | -    | -    | •    | •      | •    | •    | •    |
| Funkcije                                 | -                       | -                        | -                     | -                     | -                        | -                       | -        | -         | ?                  | •           | •    | •          | D      | •    | -    | -    | -    | -      | -    | ?    | ?    |
| Aksiomi                                  | -                       | -                        | -                     | -                     | -                        | -                       | -        | D         | ?                  | •           | •    | •          | •      | •    | -    | -    | -    | -      | -    | ?    | •    |
| Procedure                                | -                       | -                        | -                     | -                     | -                        | -                       | -        | -         | ?                  | •           | •    | •          | •      | •    | -    | -    | -    | -      | -    | -    | D    |
| Prod. pravila                            | -                       | -                        | -                     | -                     | -                        | -                       | -        | -         | ?                  | •           | •    | -          | -      | •    | -    | -    | -    | -      | -    | ?    | D    |
| Sklepanje (avt.)                         |                         |                          |                       |                       |                          |                         |          |           |                    |             |      |            |        |      |      |      |      |        |      |      |      |
| Klasifikacija                            | -                       | -                        | -                     | -                     | -                        | -                       | ?        | ?         | ?                  | •           | ?    | -          | -      | -    | -    | -    | -    | -      | -    | •    | •    |
| Dedovanje                                | -                       | -                        | -                     | -                     | -                        | -                       | ?        | ?         | ?                  | •           | •    | •          | •      | •    | -    | -    | -    | ?      | -    | •    | •    |
| Izvajanje proc.                          | -                       | -                        | -                     | -                     | -                        | -                       | -        | -         | ?                  | •           | ?    | •          | •      | •    | -    | -    | -    | -      | -    | -    | ?    |
| Veriženje                                | -                       | -                        | -                     | -                     | -                        | -                       | -        | -         | ?                  | •           | ?    | -          | •      | •    | -    | -    | -    | -      | -    | -    | D    |
| Orodje obstaja                           | -                       | -                        | -                     | -                     | -                        | -                       | ?        | ?         | ?                  | •           | ?    | -          | •      | •    | -    | -    | -    | -      | -    | •    | •    |
| Spletna kompat.                          |                         |                          |                       |                       |                          |                         |          |           |                    |             |      |            |        |      |      |      |      |        |      |      |      |
| HTML                                     | -                       | -                        | -                     | -                     | -                        | -                       | -        | -         | -                  | -           | -    | -          | -      | -    | •    | •    | •    | •      | •    | •    | •    |
| XML                                      | -                       | -                        | -                     | -                     | -                        | -                       | -        | -         | -                  | -           | -    | -          | -      | -    | -    | •    | •    | •      | •    | •    | •    |
| Tip PZ                                   |                         |                          |                       |                       |                          |                         |          |           |                    |             |      |            |        |      |      |      |      |        |      |      |      |
| Neformal.tekst                           | •                       | •                        | •                     | •                     | •                        | •                       | •        | •         | •                  | •           | •    | •          | •      | •    | •    | •    | •    | •      | •    | •    | •    |
| Slovar                                   | -                       | -                        | •                     | -                     | -                        | •                       | •        | •         | •                  | •           | •    | •          | •      | •    | -    | -    | •    | •      | •    | •    | •    |
| Taksonomija                              | -                       | -                        | -                     | •                     | •                        | -                       | •        | •         | •                  | •           | •    | •          | •      | •    | -    | -    | •    | •      | •    | •    | •    |
| Tezaver                                  | -                       | •                        | -                     | -                     | -                        | •                       | •        | •         | •                  | •           | •    | •          | •      | •    | -    | -    | •    | •      | •    | •    | •    |
| Težkokat.onto.                           | -                       | -                        | -                     | -                     | -                        | -                       | -        | D         | •                  | •           | •    | •          | •      | •    | -    | -    | -    | -      | -    | •    | •    |

Vir: lasten. Tabela je delno povzeta po Corcho in Gomez, 2000.

Jezik OWL ima veliko uporabnih lastnosti, ki so podrobneje predstavljene v naslednjem poglavju in zaradi katerih je bil OWL-DL izbran kot najprimernejši jezik za zapis znanja o ŽS.

## 2.3. OWL – WEB ONTOLOGY LANGUAGE

V tem poglavju je predstavljen jezik OWL, ki je bil v prejšnjem podpoglavju izbran kot najprimernejši jezik za potrebe izgradnje ontologije ŽS. Opisan bo razvoj jezika, tri različice in ena od razširitev jezika, glavni konstrukti jezika, delo z jezikom, akronim OWL ter potencial jezika.

### 2.3.1. Opis in razvoj

OWL (Web Ontology Language) je družina jezikov za predstavitev znanja oz. gradnjo ontologij, ki je bil razvit pod okriljem W3C (World Wide Web Consortium). Jeziki

temeljijo na dveh semantikah, ki sta večinoma kompatibilni, vendar ne povsem. Tako OWL DL in OWL Lite temeljita na deskriptivnih logikah (Horrocks in Patel-Schneider, 2003), ki imajo lepe, zaželeno ter dobro poznane računske lastnosti. Po drugi strani pa uporablja OWL-Full rahlo neobičajen semantični model, katerega namen je zagotoviti kompatibilnost z RDFS. Sintaksa jezika OWL temelji na RDF in/ali XML, podobno kot pri jeziku DAML+OIL, katerega naslednik je OWL. OWL predstavlja eno od temeljnih tehnologij, na katerih temelji semantični splet in je do sedaj pritegnil pozornost tako akademske kot komercialne javnosti.

Razvoj jezika se je pričel leta 2001, prvi osnutki sintakse so bili na voljo leta 2002, splošno priporočilo s strani W3C pa je OWL postal leta 2004, ko je bila delovna skupina tudi uradno razpuščena. Leta 2007 je bila pod okriljem W3C ustanovljena nova delovna skupina (W3C, 2008). Do trenutka pisanja pričujočega dela, to je jesen 2008, je bilo izdelanih osem novih osnutkov za OWL verzije 2.0, ki že dobiva podporo v semantičnih urejevalnikih in sklepalnikih.

### **2.3.2. Različice jezika OWL ter razširitev**

Obstajajo tri različice jezika OWL, pri čemer ima vsaka od njih svoje prednosti in slabosti oz. različno razmerje med izraznostjo in zagotavljanjem sklepalnih mehanizmov:

*OWL Lite* je bil v osnovi namenjen podpori tistih uporabnikov, ki bi potrebovali le klasifikacijsko hierarhijo in enostavne omejitve. Tako omogoča npr. omejevanje kardinalnosti, vendar dopušča le vrednosti 0 in 1. Razvijalci so se nadejali, da bo za OWL Lite enostavneje razviti različna podporna orodja, kot za izraznejše različice jezika OWL. V praksi pa se je pokazalo, da večina podprtih omejitev omogoča le malo več kot preverjanje sintaktičnih nepravilnosti. Po drugi strani pa je možno večino konstruktov v različici OWL DL zgraditi s kombiniranjem konstruktov različice OWL Lite, zaradi česar se je razvoj orodij za OWL Lite izkazal za enako težavnega kot razvoj orodij za OWL DL. Zaradi vsega povedanega se danes OWL Lite le redko uporablja.

*OWL DL* je bil narejen z namenom, da priskrbi maksimalno možno izraznost, pri čemer pa ohrani računsko popolnost (vsi zaključki so izračunljivi), rešljivost (vsi izračuni bodo izvedeni v realnem času) ter razpoložljivost uporabnih sklepalnih mehanizmov. OWL DL uporablja vse konstrukte jezika OWL, vendar pod določenimi pogoji. Tako npr. šteвне omejitve (angl. number restrictions) ne smejo biti postavljene za lastnosti, ki so definirane kot tranzitivne.

*OWL Full* temelji na drugačni semantiki kot OWL Lite ali OWL DL in je bil razvit z namenom, da bi ohranil kompatibilnost z RDFS. Tako lahko npr. istočasno razumemo razred (angl. class) v OWL Full kot skupek posameznikov (ang. individuals) ali kot posameznika samega po sebi. To seveda ni možno v OWL DL. OWL Full omogoča, da ontologija poveča pomen vnaprej definiranega (RDF ali OWL) slovarja. Zaradi tega niti v teoriji ne obstaja orodje, ki bi podpiralo popolno sklepanje nad znanjem, zapisanim v OWL Full.

Vsaka od različic je sintaktična razširitev enostavnejšega predhodnika, zato veljajo naslednje izjave, inverzne izjave pa ne:

- Vsaka veljavna OWL Lite ontologija je veljavna OWL DL ontologija.
- Vsaka veljavna OWL DL ontologija je veljavna OWL Full ontologija.

- Vsak resničen OWL Lite sklep je resničen OWL DL sklep.
- Vsak resničen OWL DL sklep je resničen OWL Full sklep.

Ker pa nobena od različic ne omogoča izpeljave pravil, je bil razvit jezik SWRL (Semantic Web Rule Language) (Horrocks et al., 2004), ki razširja OWL, kot OCL razširja UML.

### 2.3.3. Konstrukti jezika OWL

V tem delu bo predstavljen pregled konstruktov jezika OWL, ki je povzet po McGuinness in Harmelen (2003). Predpona `rdf:` ali `rdfs:` označuje, da je pojem z istim imenom in pomenom definiran že v RDF ali RDFS. Ostali pojmi so prvič definirani v OWL in bi jih morali natančneje navesti s predpono `owl:`, vendar predpono zaradi preglednosti izpustimo.

Konstrukti jezika OWL Lite:

#### Posebnosti RDFS:

`Class` (Thing, Nothing)  
`rdfs:subClassOf`  
`rdf:Property`  
`rdfs:subPropertyOf`  
`rdfs:domain`  
`rdfs:range`  
`Individual`

#### (Ne)enakosti:

`equivalentClass`  
`equivalentProperty`  
`sameAs`  
`differentFrom`  
`AllDifferent`  
`distinctMembers`

#### Karakteristike lastnosti:

`ObjectProperty`  
`DatatypeProperty`  
`inverseOf`  
`TransitiveProperty`  
`SymmetricProperty`  
`FunctionalProperty`  
`InverseFunctionalProperty`

#### Omejitve lastnosti:

`Restriction`  
`onProperty`  
`allValuesFrom`  
`someValuesFrom`

#### Omejitve kardinalnosti:

`minCardinality` (only 0 or 1)  
`maxCardinality` (only 0 or 1)  
`cardinality` (only 0 or 1)

#### Osnovne informacije:

`Ontology`  
`imports`

#### Prekrivanje razredov:

`intersectionOf`

#### Verzioriranje:

`versionInfo`  
`priorVersion`  
`backwardCompatibleWith`  
`incompatibleWith`  
`DeprecatedClass`  
`DeprecatedProperty`

#### Lastnosti beležk:

`rdfs:label`  
`rdfs:comment`  
`rdfs:seeAlso`  
`rdfs:isDefinedBy`  
`AnnotationProperty`  
`OntologyProperty`

#### Tip podatkov:

`xsd datatypes`

Konstrukti jezikov OWL DL in OWL Full vključujejo poleg konstruktov jezika OWL Lite še:

#### Aksiomi razredov:

`oneOf`, `dataRange`  
`disjointWith`  
`equivalentClass`  
 (applied to class expressions)  
`rdfs:subClassOf`

#### Booleanove kombinacije nad razredi:

`unionOf`  
`complementOf`  
`intersectionOf`

(applied to class expressions)

**Kardinalnosti:**

minCardinality  
maxCardinality  
cardinality

**Polnilne informacije:**

hasValue

V naslednjem podpoglavju si bomo ogledali dva pomembnejša konstrukta, `Class` in `Individual`, da dobimo občutek, kako se dela s konstrukti, kakšna sta njihov namen in uporabnost.

### 2.3.4. *Konstrukt class*

Konstrukt `Class` (slov. razred) definira skupino posameznikov (angl. individuals), ki spadajo skupaj, ker imajo nekaj skupnih lastnosti (McGuinness in Harmelen, 2003). Janez in Marija sta npr. oba člana razreda `Oseba`. Razrede lahko organiziramo v posebno hierarhijo s pomočjo konstrukta `subClassOf` (slov. podrazred). Vsak osnovni koncept nekega področja proučevanja naj bi imel pripadajoč razred, ki je začetek novega taksonomičnega drevesa. Obstaja vnaprej definiran in najbolj splošen razred z imenom `Thing`, ki je razred vseh posameznikov in je nadrazred vseh razredov OWL. Z drugo besedo, vsak uporabniško definiran razred je podrazred razreda `owl:Thing`. Podobno je predefiniran najbolj specifičen razred z imenom `Nothing`, ki je razred brez primerkov in je obenem podrazred vsakega razreda OWL.

Sedaj si bomo na primeru ontologije vin (SSKJ: Vino – alkoholna pijača iz soka grozdja po alkoholnem vrenju) pogledali, kako se uporablja definiranje razredov, podrazredov in sklicevanje nanje. Ta prikaz je povzet po Smith et al. (2004).

Najprej ustvarimo tri vrhnje razrede: `Winery`, `Region` in `ConsumableThing`.

```
<owl:Class rdf:ID="Winery"/>
<owl:Class rdf:ID="Region"/>
<owl:Class rdf:ID="ConsumableThing"/>
```

Omenimo, da smo do sedaj zapisali le, da obstajajo trije razredi z določenimi imeni, ki smo jih določili s sintakso `rdf:ID=`. Formalno ne vemo o teh razredih skoraj ničesar, razen tega, da obstajajo in to kljub temu, da smo jim dali znana imena (čeprav v angleščini). Lahko bi jim dali imena `razred1`, `razred2` in `razred3`. Poleg tega, da ne poznamo nobenih lastnosti definiranih razredov, razen imen, ti razredi nimajo nobenih članov.

S sintakso `rdf:ID="Region"` smo uvedli neko ime kot del definicije. Zato se lahko v nadaljevanju dokumenta sklicujemo na razred `Region` z ukazom `#Region`, kot v `rdf:resource="#Region"`. Iz drugih dokumentov ali ontologij se na ta razred lahko vedno sklicujemo prek naslova `"http://www.w3.org/TR/2004/REC-owl-guide-20040210/wine#Region"`, ker je ontologija vin dostopna prek spleta oz. shranjena na tem naslovu. V praksi si ostale ontologije definirajo svoje pojme, pri čemer so definicije zapisane v prvi ontologiji:

```

...
<!ENTITY vin "http://www.w3.org/TR/2004/REC-owl-guide-
20040210/wine#" >
...
<rdf:RDF xmlns:vin ="http://www.w3.org/TR/2004/REC-owl-guide-
20040210/wine#"
...

```

V prvem primeru smo definirali pojem vin v XML in v drugem primeru v sintaksi RDF. Na razred `Region` bi se nato lahko sklicevali v sintaksi XML z `vin:Winery` ali prek vrednosti atributa `&vin;Winery`.

Neka druga oblika sklicevanja uporablja sintakso `rdf:about="#Region"`, da razširi definicijo izvora. Uporaba te sintakse je ključni element pri gradnji porazdeljenih ontologij. Dovoljuje namreč razširjanje uvoženih definicij brez spreminjanja originalnih definicij oz. dokumenta, kjer je podana originalna definicija. Šele na tak način je omogočena postopna gradnja večjih ontologij.

Osnovni konstruktor za gradnjo taksonomij razredov je `rdfs:subClassOf`. Vzpostavlja povezavo med bolj specifičnim in bolj splošnim razredom. Če je X podrazred razreda Y, potem je vsak primerek razreda X obenem primerek razreda Y. Relacija `rdfs:subClassOf` je tranzitivna, kar pomeni, da če je X podrazred Y in je Y podrazred razreda Z, potem je X podrazred razreda Z. Tako lahko definiramo `PotableLiquid` (tekočine, primerne za pitje) kot podrazred razreda `ConsumableThing` (užitne stvari):

```

<owl:Class rdf:ID="PotableLiquid">
  <rdfs:subClassOf rdf:resource="#ConsumableThing" />
  ...
</owl:Class>

```

V primeru spletnih ontologij bi bila lahko oba razreda definirana v ločenih ontologijah, ki bi zagotavljali osnovne bloke za različne ontologije hrane ali pijač, kar je bilo tudi storjeno. Ontologija hrane<sup>26</sup> je bila uvožena v ontologijo vin. Ontologija hrane vsebuje veliko razredov, kot so `Food`, `EdibleThing`, `MealCourse` in `Shellfish`, ki sicer ne spadajo neposredno na področje vin, jih pa potrebujemo, če želimo izvajati smiselno sklepanje nad znanjem, ki ga bomo zapisali v ontologijo vin. Navsezadnje sta področji hrane in vin zelo povezani.

Definicija razreda ima dva dela, pri čemer je prvi del definicija imena, drugi pa lista omejitev. Vsaka od naslednjih izjav v definiciji razreda nadalje omejuje primerke definirane razreda. Primerki razreda spadajo v presek omejitev. Do sedaj smo pokazali le primere z eno omejitvijo, ki je ukazovala, da je nek razred podrazred nekega drugega razreda. Na tem mestu lahko sedaj ustvarimo še enostavno definicijo razreda `Wine`. `Wine` je `PotableLiquid`. Definirajmo še `Pasta` kot `EdibleThing`:

```

<owl:Class rdf:ID="Wine">
  <rdfs:subClassOf rdf:resource="#&food;PotableLiquid"/>
  <rdfs:label xml:lang="en">wine</rdfs:label>
  <rdfs:label xml:lang="fr">vin</rdfs:label>

```

<sup>26</sup> <http://www.w3.org/TR/2004/REC-owl-guide-20040210/food.rdf>

```

...
</owl:Class>

<owl:Class rdf:ID="Pasta">
  <rdfs:subClassOf rdf:resource="#EdibleThing" />
  ...
</owl:Class>

```

Z zapisom `rdfs:label` lahko definiramo opcijska imena za določen razred, ki se jih lahko kasneje uporabi pri predstavitvi ali ko z ontologijo delajo ljudje. Z zgornjim zapisom smo definirali angleško besedo `wine` in francosko besedo `vin` za razred `Wine`. Atribut `lang` zagotavlja večjezično podporo. `Label` je v bistvu komentar in v ničemer ne vpliva na logično interpretacijo ontologije.

Naša ontologija `vin` je še vedno zelo nepopolna, saj ne vemo skoraj ničesar o vinih, razen da so to primerki razreda `Thing` in `PotableLiquid`, vendar imamo dovolj informacij, da lahko definiramo posameznike ter izvajamo nekaj sklepanja o njih.

### 2.3.5. *Konstrukt Individual*

Poleg razredov, ki so bili predstavljeni v prejšnjem podpoglavju, želimo običajno opisati tudi njihove člane (McGuinness in Harmelen, 2003). Običajno razumemo člane razredov kot posameznike v našem svetu. Posamezniki so primerki razredov, njihovi atributi se lahko uporabijo za vzpostavitev povezav med njimi. Tako lahko opišemo nekega Janeza kot primerek razreda `Person` in uporabimo atribut `hasEmployer`, da povežemo posameznika `Janez` in posameznika `UniverzavLjubljani`.

Posameznik je definiran že s tem, ko določimo, katerega razreda član je (Smith et al., 2004). Če se vrnemo na naš primer ontologije `vin`, lahko zapišemo:

```
<Region rdf:ID="CentralCoastRegion" />
```

Z eno izjavo smo definirali, da je `CentralCoastRegion`, ki je neka določena pokrajina, član razreda `Region`. Isto informacijo bi lahko zapisali tudi na daljši način:

```

<owl:Thing rdf:ID="CentralCoastRegion" />

<owl:Thing rdf:about="#CentralCoastRegion">
  <rdfs:type rdf:resource="#Region"/>
</owl:Thing>

```

Pri tem je treba opomniti, da je `rdfs:type` atribut RDF, ki povezuje posameznika z razredom, katerega član je. V tem drugem načinu, ko smo informacijo dobili prek dveh izjav, ni nobene potrebe, da sta obe izjavi v dokumentu postavljeni ena ob drugo, niti ni potrebno, da sta sploh vključeni v isti dokument oz. ontologijo. Spletne ontologije naj bi bile narejene tako, da so lahko porazdeljene. Morajo pa biti v tem primeru seveda povezane prek naslovov URL.

### 2.3.6. *Domneva odprtega sveta*

Pomembna lastnost jezika OWL je, da uporablja »domnevo odprtega sveta«, kar pomeni, da če z uporabo trenutnega znanja ne moremo dokazati, da je izjava resnična, to še ne



pomeni, da je izjava neresnična. Po domače bi rekli, da se zavedamo, da nikoli ne vemo vsega. Ta lastnost je nujni pogoj, če želimo uporabiti OWL za zapis znanja, ki ga nameravamo sproti dopolnjevati, kot je to v primeru semantičnega spleta. Ta lastnost tudi predstavlja eno od pomembnih razlik med OWL in recimo jezikoma SQL ali Prolog.

### **2.3.7. Delo z jezikom OWL**

Obstoječe ontologije OWL lahko najdemo na različnih spletnih straneh, kot je stran urejevalnika Protege<sup>27</sup> ali stran owl-ontologies.com<sup>28</sup>. Lahko jih poiščemo s kakšnim od spletnih brskalnikov, kjer iščemo datoteke s končnico .owl. Lahko pa uporabimo kar semantično različico brskalnika Google z imenom Swoogle<sup>29</sup>, ki išče kar po ontologijah samih. O orodjih za delo z OWL na tem mestu ne bomo govorili, ker je temu namenjeno celotno poglavje 2.4.2.

### **2.3.8. Akronim OWL**

Pričakovan akronim za jezik Web Ontology Language bi bil WOL in ne OWL. Pa vendar je bil kot akronim izbran OWL, ker je enostavnejši za izgovorjavo, asociira na sovo, simbol modrosti, logično sklepanje ter spominja na projekt iz sedemdesetih let prejšnjega stoletja One World Language. Poleg tega avtorji jezika v šali opravičujejo nekonsistentnost pri akronimu z besedami: »Zakaj ne bi bili nekonsistentni vsaj v enem vidiku jezika, ki se ves čas ukvarja s konsistentnostjo.«

### **2.3.9. Potencial jezika OWL**

Eno najbolj obetajočih področij za uporabo jezika OWL je semantični splet, ki zbuja visoka pričakovanja, ker predstavlja okvir, ki naj bi omogočil izmenjavo, razumevanje ter uporabo informacij preko meja različnih aplikacij, organizacij in skupnosti. Je plod sodelovanja mnogih raziskovalcev in podjetij, ki ga vodi W3C. Glavni bloki, na katerih je osnovan, so RDF, OWL ter SPARQL (SPARQL Protocol and RDF Query Language) in GRDDL (Gleaning Resource Descriptions from Dialects of Languages). Tim Berners-Lee (2008), znan kot izumitelj spleta in eden vodilnih pri W3C, je v objavljenem intervjuju zatrdil, da so vsi gradniki za izgradnjo semantičnega spleta že dostopni. Ker pa semantični splet kar ne zaživi, kot bi po mnenju predlagateljev lahko, se odpira vprašanje, kaj je razlog tega (Why, 2008).

## **2.4. UREJEVALNIKI ONTOLOGIJ**

Urejevalniki ontologij so orodja, s katerimi si pomagamo pri urejanju ontologij oz. znanja, zapisanega v ontologijah. Ker bomo pri delu potrebovali tako orodje, je v tem podpoglavju najprej nekaj na splošno povedanega o urejevalnikih, nato so predstavljena orodja za delo z jezikom OWL, ki je bil predstavljen v prejšnjem podpoglavju, na koncu pa bo predstavljeno orodje Protege-OWL, ki ga bomo pozneje uporabili za gradnjo ontologije ŽS.

---

<sup>27</sup> [http://protegewiki.stanford.edu/index.php/Protege\\_Ontology\\_Library](http://protegewiki.stanford.edu/index.php/Protege_Ontology_Library)

<sup>28</sup> <http://www.owl-ontologies.com/index.html>

<sup>29</sup> <http://swoogle.umbc.edu/>

### 2.4.1. Splošno o urejevalnikih

Orodja za delo z ontologijami se ločijo po jeziku, ki ga podpirajo, in po funkcionalnosti, ki jo omogočajo. Če jih ločimo po funkcionalnosti, imamo v grobem sedem skupin orodij (Gomez et al., 2004):

- Orodja za gradnjo ontologij se uporabljajo za gradnjo ontologij od začetka. Poleg splošnih funkcij za urejanje in iskanje pomagajo ta orodja pri dokumentiranju, izvažanju ter uvažanju iz drugih formatov in jezikov, omogočajo grafične predstavitve itd. Delimo jih še glede na to, ali so narejena za en sam jezik ali jih podpirajo več. V prvo skupino spadajo Ontolingua Server za Ontolingua in KIF, OntoSaurus za Loom, WebOnto za OCM ter OilEd za OIL in DAML-OIL. V drugo skupino pa spadajo Protege, WebODE, OntoEdit in KAON. Slednja orodja imajo svoj model predstavitve znanja, ki običajno omogoča enostavno dodajanje funkcionalnosti in razširitev.
- Orodja za preverjanje ontologij se uporabljajo, kadar želimo integrirati in uporabljati ontologije v ostalih informacijskih sistemih. Pred tem je namreč treba preveriti vsebino same ontologije.
- Orodja za združevanje in povezovanje ontologij. V to skupino spadajo posamezni moduli zgoraj omenjenih orodij, kot je vtičnik PROMPT za Protege, ki pa je v zadnji verziji že sestavni del Protegeja. Obstajajo pa tudi posebej za ta namen zgrajena okolja, kot sta WebODE<sup>30</sup> in Ontolingua Server<sup>31</sup>, ki predstavlja cel set orodij za pomoč pri gradnji porazdeljenih ontologij in ob hkratnem sodelovanju različnih skupin razvijalcev. Server vsebuje knjižnico ontologij, prevajalnikov med različnimi jeziki (Prolog, CORBA IDL, CLIPS, Loom ...) ter urejevalnikov.
- Na ontologijah temelječa orodja za označevanje so se pojavila s prihodom semantičnega spleta in se uporabljajo za (pol)avtomatično podajanje znanja iz ontologije na splet. Sem spada tudi različica beta programa Swoogle.
- Orodja za poizvedovanje po ontologijah so običajno močno odvisna od jezika ontologije.
- Orodja za učenje ontologij lahko (pol)avtomatično gradijo ontologije iz tekstov v naravnem jeziku kot tudi iz polstrukturiranih virov in baz podatkov.
- Tako imenovani ontološki strežniki (angl. ontology servers). Ahmad in Colomb (2007) jih definirata kot informacijski sistem, ki podpira ontologije. Avtorja jih primerjata med seboj ter ugotavljata, da so vsi še zelo na začetni razvojni stopnji.

### 2.4.2. Orodja za OWL

Ker smo v podpoglavju o tehnikah predstavitve znanja (2.2.4) izbrali jezik OWL kot najprimernejšega za naše potrebe, nas zanimajo predvsem orodja, ki podpirajo ta jezik.

Za urejanje, gradnjo novih ali pregledovanje obstoječih ontologij, zapisanih z OWL, uporabljamo urejevalnike ontologij, (Protege<sup>32</sup>, OilEd, pOWL<sup>33</sup>, SWOOP<sup>34</sup>...). Za skladnost zapisa v OWL in morebitne napake se uporabljajo že v urejevalnike vgrajena

---

<sup>30</sup> <http://delicias.dia.fi.upm.es/webODE/>

<sup>31</sup> <http://ontolingua.stanford.edu/>

<sup>32</sup> <http://protege.stanford.edu/>

<sup>33</sup> <http://sourceforge.net/projects/powl>

<sup>34</sup> <http://www.mindswap.org/2004/SWOOP/>

orodja ali samostojna orodja, kot je OWL Ontology Validator<sup>35</sup>. Za urejanje posebnih ontologij, zapisanih z OWL, uporabljamo specialna orodja, kot je v primeru OWL-S orodje Owl-S Editor<sup>36</sup>.

Med orodja za izvajanje sklepanja spadajo Pellet<sup>37</sup>, FaCT++<sup>38</sup> in Racer<sup>39</sup>. Za delo z jezikom SWRL, ki razširja OWL, kot OCL razširja UML, in ki omogoča izpeljavo pravil, se uporabljata Hoolet<sup>40</sup> in Bossam<sup>41</sup>. Hoolet deluje tako, da prevede zapis OWL v logiko prvega reda in potem izvede sklepanje. Bossam je v bistvu semantični sklepalnik, ki temelji na pravilih RETE in podpira sklepanje nad zapisi OWL, SWRL in RuleML.

Obstaja tudi veliko vtičnikov za omenjene jezike oz. za urejevalnike ontologij, temelječih na teh jezikih. Za Protege je teh vtičnikov približno sto, ker je Protege odprtokodni projekt, kar omogoča aktivno sodelovanje različnih skupnosti uporabnikov pri dodajanju novih funkcionalnosti.

Pri izbiri najprimernejšega orodja je treba biti pozoren na:

- interoperabilnost z drugimi orodji ter na možnost uvoza in izvoza ontologij, zapisanih z različnimi jeziki;
- izraznost uporabljenega modela predstavitve znanja;
- možnost razširitev in nadgradenj;
- dostopnost in sposobnost sklepalnih orodij;
- uporabnost vgrajenih orodij.

Do sedaj je bilo opravljenih že več vzporednih primerjav in testiranj med različnimi urejevalniki ontologij, ki pa so vse že malo zastarele. Kakorkoli, Duineveld et al. (2000) so primerjali med seboj Ontolingua, WebOnto, ProtégéWin, OntoSaurus, ODE in KADS22. Na podlagi triindvajsetih kriterijev so prišli do zaključka, da spada Protege med orodja, ki jih je najlažje uporabljati. Podobno so Lambrix et al. (2003) primerjali Protege-2000, Chimaera, DAG-Edit in OilEd na področju bioinformatike. Na podlagi petnajstih kriterijev so zaključili, da ima Protege prednost pred drugimi orodji predvsem na področju uporabniškega vmesnika, možnosti dodajanja funkcionalnosti prek novih vtičnikov, že obstoječe funkcionalnosti ter števila podprtih jezikov za predstavitev znanja.

Na podlagi povedanega smo se odločili, da za urejanje ontologije ŽS tudi sami uporabimo urejevalnik ontologij Protege.

### **2.4.3. PROTEGE-OWL – urejevalnik za OWL**

Protege<sup>42</sup> je prosto dostopna, odprtokodna platforma, namenjena urejanju ontologij. Zaradi odprtokodnosti in dostopnosti ima široko bazo uporabnikov, ki prispevajo veliko novih orodij v obliki vtičnikov (angl. plug-in) ali javanskih vmesnikov (angl. Java-based APIs) na področjih vizualizacije, kreiranja ontologij, povezovanja med njimi itd., ki olajšujejo

<sup>35</sup> <http://www.mygrid.org.uk/OWL/Validator>

<sup>36</sup> <http://staff.um.edu.mt/cabe2/supervising/undergraduate/owlSeditFYP/OwlSEdit.html>

<sup>37</sup> <http://pellet.owldl.com/>

<sup>38</sup> <http://owl.man.ac.uk/factplusplus/>

<sup>39</sup> <http://www.racer-systems.com/products/tools/protege.phtml>

<sup>40</sup> <http://owl.man.ac.uk/hoolet>

<sup>41</sup> <http://en.wikipedia.org/wiki/Bossam>

<sup>42</sup> <http://protege.stanford.edu/>

delo z urejevalnikom. Protege uporablja lasten model predstavitve znanja in omogoča zapis znanja z različnimi jeziki.

Dva glavna načina zapisa znanja v Protegeju sta okvirji in jezik OWL. Ker smo v enem od prejšnjih poglavij izbrali OWL kot najprimernejši format za zapis naše ontologije, se v nadaljevanju osredotočam le na ta format. Format zapisa OWL je za Protege dostopen kot vtičnik Protege-OWL (Knublauch et al., 2004). Omogoča gradnjo ontologij OWL in RDF, urejanje in vizualizacijo razredov, atributov in pravil SWRL, zapis lastnosti razredov v obliki logičnih izjav v OWL, poganjanje sklepalnikov ter urejanje primerkov OWL za potrebe semantičnega spleta.

## **2.5. METODOLOGIJE ZA GRADNJO ONTOLOGIJ**

Po namenu ločimo metodologije na take za izgradnjo ontologij od začetka, za predelavo in dopolnjevanje ontologij, za zajem znanja, za združevanje in povezovanje ontologij, metodologije za podporo porazdeljene izgradnje ontologij ter metodologije za preverjanje oz. vrednotenje ontologij. Ker bomo gradili ontologijo ŽS od začetka, nas zanimajo predvsem metodologije, ki so temu namenjene. V tem podpoglavju bo najprej opravljena primerjava metodologij za gradnjo ontologij od začetka, nato pa bo predstavljena metodologija METHONTOLOGY, po kateri bomo v nadaljevanju gradili ontologijo ŽS.

### **2.5.1. Primerjava metodologij za gradnjo ontologij**

Pojem »proces izgradnje ontologij« se je pojavil šele leta 1997 in temelji na standardih IEEE<sup>43</sup> za razvoj programske opreme.

Obstaja več metodologij za gradnjo ontologij glede na kompleksnost ciljne ontologije. Tako nekateri pristopi h gradnji lahkih ontologij zajemajo diagramске tehnike UML ter pristope za gradnjo baz podatkov, npr. modele ER. V zadnjih dveh desetletjih se je pojavilo več formalnih metodologij, katerih namen je izgradnja težkokategornih ontologij in ki vključujejo tudi pristope s področja umetne inteligence.

Eden od pomembnih vidikov pri gradnji ontologij je strukturiranje znanja, za katerega obstajajo načeloma trije različni načini: formaliziranje lahko poteka od zgoraj navzdol, od spodaj navzgor ali pa iz sredine v obe smeri. Pri formalizaciji od zgoraj navzdol začnemo z najbolj splošnimi pojmi in jih rafiniramo toliko časa, da pridemo do konceptov, ki jih potrebujemo za določeno nalogo. Pri tem pristopu je nevarno, da na višjih nivojih formaliziramo znanje, ki ga pri reševanju določene naloge ne bomo nikoli potrebovali, oz. da bo formalizacija znanja preširoka in presplošna. Obratno velja za formalizacijo od spodaj navzgor, kjer začnemo z najbolj specifičnimi koncepti in potem nadaljujemo z bolj splošnimi. Zaradi osredotočenja na podrobnosti pa se pri takem pristopu včasih spregleda celotno sliko, koncepti na višjih nivojih so preslabo razdelani ali kakšen tudi manjka. Pomanjkljivosti omenjenih pristopov odpravlja pristop iz sredine navzven, kjer se ontologija začne graditi pri osrednjih konceptih, nato pa se generalizira navzgor in specializira navzdol. Tak pristop zagotavlja ravnotežje med preveliko splošnostjo prvega in preveliko specifičnostjo drugega (Berčič, 2006) (Gomez-Perez, 2003, str. 11).

---

<sup>43</sup> <http://www.ieee.org>

Med metodologije za izgradnjo ontologij od začetka spadajo metodologije Cyc, Uschold in Kingova metodologija, Gruninger in Foxova metodologija, pristop KACTUS, metoda SENSUS, metodologija On-To-Knowledge in METHONTOLOGY (Gomez et al., 2004). Njihova primerjava je predstavljena v spodnji tabeli, pri čemer »Opisano« pomeni, da metodologija opisuje, kaj je treba storiti v posameznem koraku, kdaj in kdo naj to naredi, »Predlagano« pomeni, da metodologija le identificira določen korak in »-« označuje, da v uradni dokumentaciji določen korak ni omenjen. Za podrobnejšo razlago ter opis okvirja za primerjavo naj si bralec pogleda Gomez et al. (2004).

Tabela 4: Pregled lastnosti metodologij za izgradnjo ontologij.

|                                          |                                       | Cyc                          | Uschold & King  | Gruninger & Fox    | KACTUS             | METHONTOLOGY                                                     | SENSUS            | On-To-Knowledge     |
|------------------------------------------|---------------------------------------|------------------------------|-----------------|--------------------|--------------------|------------------------------------------------------------------|-------------------|---------------------|
| <b>Podpora življenjskemu ciklu</b>       |                                       | Razvojni prototipi           | Ne podpira      | Razvojni prototipi | Razvojni prototipi | Razvojni prototipi                                               | Ne podpira        | Razvojni prototipi  |
| <b>Odvisnost od aplikacij</b>            |                                       | Neodvisna                    | Neodvisna       | Delno odvisna      | Odvisna            | Neodvisna                                                        | Delno             | Odvisna             |
| <b>Identifikacija konceptov</b>          |                                       | Ni določeno                  | Iz sredine ven  | Iz sredine ven     | Od zgoraj dol      | Iz sredine ven                                                   | Ni določeno       | Vse možnosti        |
| <b>Orodja, ki podpirajo metodologijo</b> |                                       | Cyc orodja                   | Ni takih orodij | Ni takih orodij    | Ni takih orodij    | ODE <sup>44</sup><br>WebODE <sup>45</sup><br>OntoEdit<br>Protege | Ni takih orodij   | OntoEdit z vključki |
| <b>Menedžment ontologije</b>             | <b>Planiranje</b>                     | -                            | -               | -                  | -                  | Predlagano                                                       | -                 | Opisano             |
|                                          | <b>Kontrola</b>                       | -                            | -               | -                  | -                  | Predlagano                                                       | -                 | Opisano             |
|                                          | <b>Zagotavljanje kakovosti</b>        | -                            | -               | -                  | -                  | Predlagano                                                       | -                 | Opisano             |
| <b>Aktivnosti gradnje</b>                | <b>Predizgradnja</b>                  | <b>Analiza okolja</b>        | -               | -                  | -                  | -                                                                | -                 | Predlagano          |
|                                          |                                       | <b>Analiza izvedljivosti</b> | -               | -                  | -                  | -                                                                | -                 | Opisano             |
|                                          | <b>Gradnja</b>                        | <b>Specifikacija</b>         | -               | Predlagano         | Opisano v detajle  | Predlagano                                                       | Opisano v detajle | Opisano v detajle   |
|                                          |                                       | <b>Konceptualizacija</b>     | -               | -                  | Opisano v detajle  | Predlagano                                                       | Opisano v detajle | Predlagano          |
|                                          |                                       | <b>Formalizacija</b>         | -               | -                  | Opisano v detajle  | Opisano                                                          | Opisano           | Opisano             |
|                                          |                                       | <b>Implementacija</b>        | Predlagano      | Predlagano         | Opisano            | Predlagano                                                       | Opisano v detajle | Opisano             |
|                                          | <b>Poizgradnja</b>                    | <b>Vzdrževanja</b>           | -               | -                  | -                  | Predlagano                                                       | -                 | Predlagano          |
|                                          |                                       | <b>Uporaba</b>               | -               | -                  | -                  | -                                                                | -                 | Predlagano          |
| <b>Podpora</b>                           | <b>Zbiranje znanja</b>                |                              | Predlagano      | Predlagano         | Predlagano         | -                                                                | Opisano v detajle | Opisano             |
|                                          | <b>Vrednotenje</b>                    |                              | -               | Predlagano         | Opisano v detajle  | -                                                                | Opisano v detajle | Predlagano          |
|                                          | <b>Menedžment povezanih ontologij</b> |                              | Predlagano      | Predlagano         | Predlagano         | Predlagano                                                       | -                 | Predlagano          |
|                                          | <b>Dokumentiranje</b>                 |                              | Predlagano      | Predlagano         | Predlagano         | -                                                                | Opisano v detajle | Opisano             |

Vir: Gomez et al., 2004.

### 2.5.2. Metodologija METHONTOLOGY

Na podlagi primerjave, prikazane v zgornji tabeli (Tabela 4), smo se za potrebe pričujočega magistrskega dela odločili za uporabo metodologije METHONTOLOGY. Prvi od razlogov je, da ta metodologija pokriva največ korakov same gradnje ontologije. Drugi

<sup>44</sup> ODE (Blazquez et al., 1998)

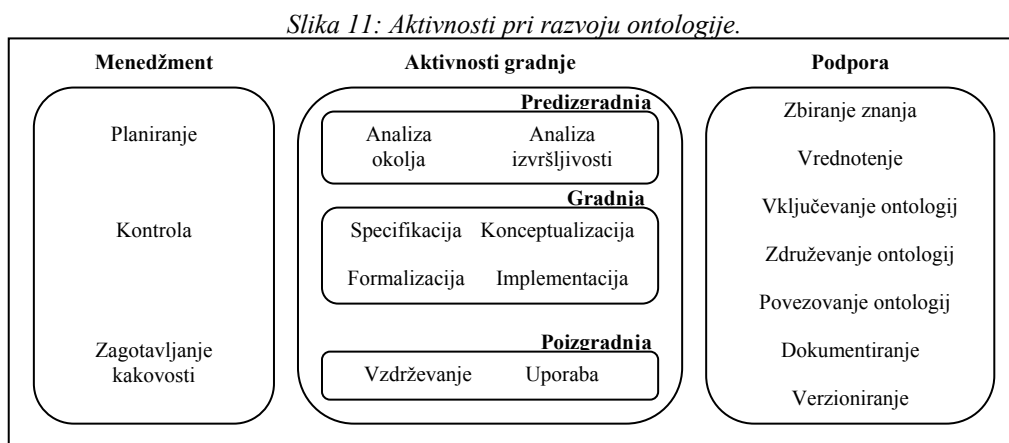
<sup>45</sup> WebODE (Arpirez et al., 2003)

razlog je močna formalizacija opisa korakov ter osredotočenost na formalno logiko, zaradi česar je bilo moč to metodologijo tudi informacijsko podpreti. Tretji razlog je, da tudi Protege, ki je bil izbran kot najprimernejši urejevalnik za jezik OWL, v veliki meri podpira to metodologijo. Četrti razlog je siceršnja sprejetost metodologije, saj je bila metodologija uporabljena v več projektih za gradnjo ontologij, kot so »Chemicals«, »Monatomic Ions«, »Silicate«, »Environmental Pollutants«, »Reference«, »OntoRoadMap«, (KA)2, KM, MKBEEM, Esperanto in druge ontologije.

### **Aktivnosti razvoja nove ontologije:**

Metodologija METHONTOLOGY vključuje identifikacijo skoraj vseh posameznih sklopov aktivnosti, ki so prikazane na Slika 11, ter življenjski krog ontologije, ki temelji na razvijanju prototipov. Vključuje tudi tehnike za izpeljavo vsake od predlaganih aktivnosti.

V METHONTOLOGY je proces izgradnje ontologije razdeljen na tri skupine aktivnosti: aktivnosti menedžmenta, same gradnje ter podporne aktivnosti, kar prikazuje spodnja slika (Slika 11) (Gomez et al., 2004).



*Vir: Gomez et al., 2004.*

Aktivnosti *menedžmenta* so pri gradnji ontologij praktično enake kot pri kakšnem drugem razvoju programske opreme. Najprej imamo *planiranje* po korakih, njihov vrstni red in porazdeljevanje sredstev, tako finančnih kot človeških. Aktivnost *kontrole* zagotavlja, da se delo odvija po načrtu, aktivnost *zagotavljanja kakovosti* pa naj bi zagotovila, da vsak izdelek (ontologija, programska oprema, dokumentacija) zadošča postavljenim standardom kakovosti.

Aktivnosti same *gradnje* so razdeljene na tri sklope: *predizgradnja*, *gradnja* in *poizgradnja*. Ti sklopi predstavljajo osrednje dejavnosti vsake gradnje ontologij. V *predizgradnji* se izdela *analiza okolja*, kjer se identificira tehnološko okolje, kjer bo ontologija uporabljena, ter aplikacije, ki jo bodo uporabljale, itd. *Analiza izvedljivosti* pa nam odgovori na vprašanje, ali je izgradnja ontologije, kot jo želimo, sploh izvedljiva in smiselna. Nato pride sklop *gradnje* in v aktivnosti *specifikacije* podrobno razčistimo, zakaj gradimo ontologijo, kaj je njen primarni namen, kakšna bo njena uporaba in kdo jo bo uporabljal. V aktivnosti *konceptualizacije* strukturiramo znanje področja v obliki lahko razumljivih modelov. V aktivnosti *formalizacije* transformiramo konceptualni model v formalnega, ki ga nato v aktivnosti *implementacije* zakodiramo v računalniku razumljiv model s pomočjo kakšnega od jezikov za predstavitev znanja, ki so bili predstavljeni v enem od prejšnjih poglavij (Poglavje 2.2). V tretjem sklopu, *poizgradnja*, pa v aktivnosti

*vzdrževanje* popravljamo in posodabljammo ontologijo, če je to potrebno. Aktivnost *uporaba* pa predstavlja cilj same izgradnje ontologije, saj se tukaj ukvarjamo s samo uporabo že implementirane ontologije oz. znanja, zapisanega v ontologiji.

Aktivnosti *podpore* se izvajajo vzporedno s samo gradnjo ontologije. Namen aktivnosti *zbiranja znanja* je ekstrakcija znanja nekega področja, kar izvedemo prek pogovorov s strokovnjaki področja, prek vprašalnikov, analize različne literature oz. dokumentacije ali prek kakšnega delno avtomatiziranega procesa. V aktivnosti *vrednotenje* izvajamo presojo ontologij, njihovih programskih okolij in dokumentacije. Aktivnost  *vključevanje ontologij* je potrebna takrat, ko gradimo ontologijo na osnovi že obstoječih ontologij. Podobno potrebujemo aktivnost *združevanje ontologij* takrat, ko iz obstoječih ontologij istega področja uporabimo znanje in ne samih ontologij. Novonastala ontologija tako običajno omogoča poenotenje in povezovanje konceptov, terminologije, konstant et alalega iz že obstoječih ontologij. V aktivnosti *povezovanje ontologij* vzpostavimo povezave med že obstoječimi ontologijami, torej jih ne spreminjamo. V aktivnosti *dokumentiranje* podrobno in popolno dokumentiramo vsako zaključeno delovno aktivnost ali izdelan rezultat. V aktivnosti *verzioniranje* pa poskrbimo, da se beležijo vse spremembe ontologije in dokumentacije, s čimer omogočimo kontrolo nad spremembami.

#### ***Aktivnost konceptualizacije:***

Ena od zelo pomembnih aktivnosti pri razvoju ontologije je konceptualizacija, kjer gre za organiziranje znanja oz. izgradnjo konceptualnega modela, ki se ga zajame v aktivnosti zajema znanja. Aktivnost konceptualizacije si zasluži posebno pozornost, ker radikalno vpliva na nadaljnji razvoj ontologije, saj v veliki meri določi samo obliko zapisanega znanja. Naloga te aktivnosti je pretvorba neformalnega znanja o nekem področju, kot ga razumejo strokovnjaki področja, v formalno obliko, ki jo strokovnjaki za gradnjo ontologij razvijejo v aktivnosti implementacije in ki je v končni fazi razumljiva in berljiva računalniškimi orodjem. Pri tem se uporabljajo tehnike, ki so v osnovi tabele ali grafiki s povezavami med koncepti. Metodologija METHONTOLOGY predlaga naslednji nabor in vrstni red korakov v aktivnosti konceptualizacije:

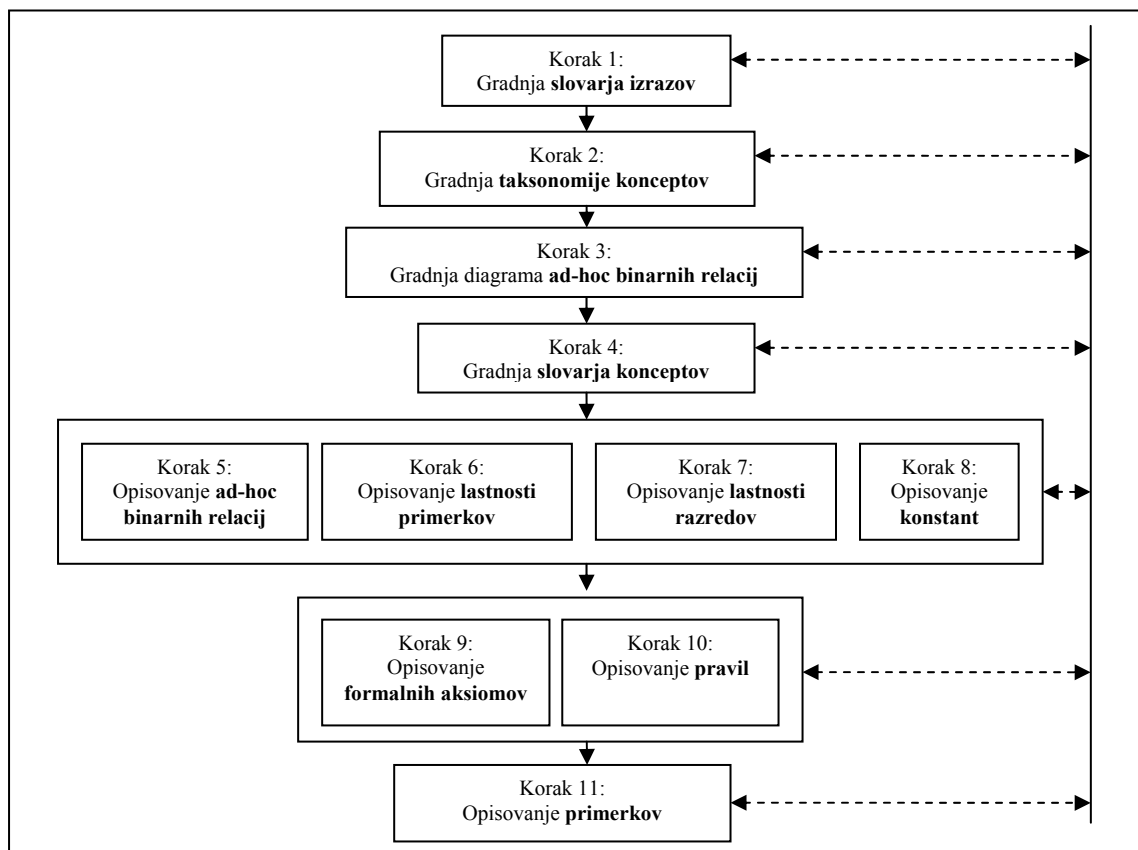
Prvi korak je izdelava neformalnega slovarja izrazov, ki so pomembni za neko področje znanja. Treba jih je zapisati, jim določiti definicije ter sinonime in akronime.

V drugem koraku se iz izrazov v slovarju izrazov definirajo koncepti, ki se jih uredi v eni ali več taksonomijah. Pri generalizaciji (angl. *superclass-of*) ali specializaciji (angl. *subclass-of*) razredov se določi, ali gre za izključujočo sestavo (angl. *disjoint-decomposition*), kjer se naštetih podrazredi nekega nadrazreda med seboj izključujejo (vsak primerek nadrazreda lahko pripada samo enemu podrazredu), ali popolno sestavo (angl. *exhaustive decomposition*), ki zahteva, da je vsak primerek nadrazreda primerek tudi vsaj enega od naštetih podrazredov. Particija (angl. *partition*) nadrazreda v podrazrede pa je kombinacija izključujoče in popolne sestave, kar pomeni, da je vsak primerek nadrazreda primerek vsaj in točno enega podrazreda v particiji. Dodatno definiranje hierarhičnih odnosov med koncepti s pomočjo particij, izključujoče in popolne sestave je eden od principov dobrega inženirstva znanja (angl. *knowledge engineering*) (Berčič, 2006).

V tretjem koraku se izdelava neformalni diagram ad-hoc binarnih relacij med koncepti naše prihodnje ontologije ter morebitnimi koncepti drugih ontologij.

V četrtem koraku se izdelata formalni slovar konceptov, ki vključuje prejšnje informacije (ime koncepta, njegova definicija, atributi in ad-hoc binarne relacije med njimi). Po potrebi se določijo metaatributi razreda. Metaatribut razreda ne opisuje lastnosti primerkov razreda, pač pa lastnosti razreda, ko je ta primerek nekega razreda višje vrste. Metaatribut razreda žival je npr. vrsta, atribut razreda oz. primerkov razreda žival pa npr. število nog.

Slika 12: Koraki znotraj aktivnosti »konceptualizacija«, kot jih predlaga metodologija METHONTOLOGY.



Vir: Gomez et al., 2004.

V petem koraku se podrobno definira vsako od ad-hoc relacij. Rezultat tega koraka je tabela ad-hoc binarnih relacij, kjer se poleg imena relacije ter konceptov, ki jih relacija povezuje, določi še nekaj metalastnosti relacij, kot so mestnost relacije (1:1, 1:N, M:N), matematične lastnosti relacije (refleksivnost, simetričnost, antisimetričnost, tranzitivnost ...), morebitno obratno relacijo (obratna relacija od prednik je npr. potomec) in druge metalastnosti.

V šestem in sedmem koraku se določi attribute primerkov. Rezultata teh dveh korakov sta tabela atributov primerkov in tabela atributov razredov, kjer se za vsak atribut (npr. starost v letih za razred človek) določi tip argumenta (npr. naravno število za starost v letih), mersko enoto (npr. leta za starost v letih), zalogo vrednosti, ki jo lahko posamezni atribut zavzame (za starost npr. od 1 do 120) ter kardinalnost lastnosti (za starost npr. 1, ker ima vsak človek samo eno starost).

V osmem koraku se zapiše glavne konstante proučevanega področja. Rezultat tega koraka je tabela konstant področja, kjer se navedejo imena konstant (npr. LetošnjeLeto), njihova vrednost (npr. 2005), tip vrednosti (npr. naravna števila) in merska enota (npr. leta).



V devetem koraku se zapišejo formalni aksiomi, ki so integritetne omejitve (integrity constraints) na podatkovnem modelu in so po definiciji resnične. Če bi jim kakšen podatek v zbirki znanja nasprotoval, bi bila zbirka znanja nekonsistentna.

V desetem koraku se določijo pravila, ki jih bomo potrebovali pri izvajanju preverjanja ter sklepanja.

Zadnji, enajsti korak je izbirni in predstavlja opisovanje ali vnašanja podatkov o primerkih. Primerki običajno niso del ontologije, pač pa konkretne zbirke znanja, vendar se včasih izkaže, da je primerno, da se kakšen primerek definira tudi v ontologiji sami.

### 3. GRADNJA ONTOLOGIJE ŽIVLJENJSKIH SITUACIJ

V tem poglavju je predstavljena gradnja ontologije ŽS. Koraki sledijo metodologiji METHONTOLOGY, ki je bila predstavljena v drugem poglavju.

#### 3.1. SPECIFIKACIJA PRIHODNJE ONTOLOGIJE

V aktivnosti *specifikacije* podrobno razčistimo, zakaj gradimo ontologijo, kaj je njen primarni namen, kakšna bo njena uporaba in kdo jo bo uporabljal.

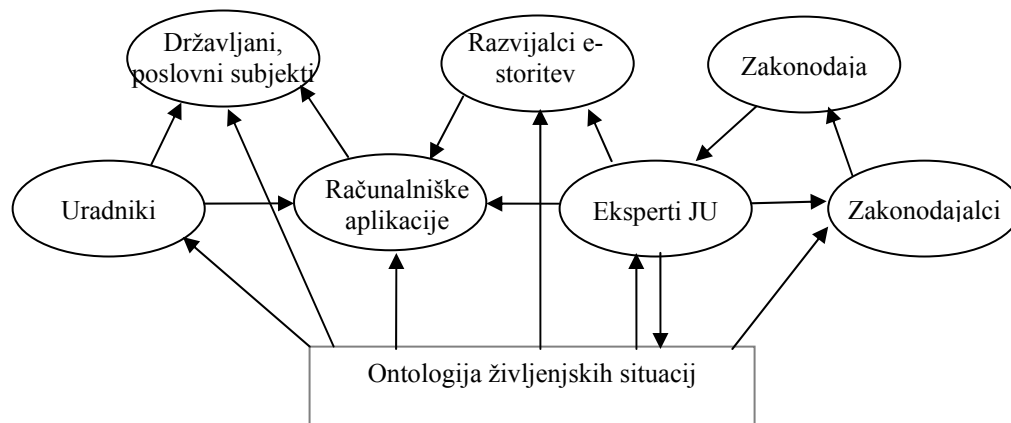
##### 3.1.1. Namen ontologije

Kot je navedeno že v prvem poglavju, je glavni namen ontologije ali modela ŽS na enem mestu zapisati vse relevantne informacije o ŽS ter za njih relevantne postopke javne uprave in dokumente, ki v njih nastopajo. Posredno je namen ontologije poenotiti terminologijo, omogočiti zapis ter dostop do znanja na enem mestu, izboljšati ažurnost informacij in preprečiti podvajanje znanja. Z vključitvijo potrebnih informacij je tretji namen ontologije zagotavljanje semantične povezljivosti e-storitev, s čimer se nadamo, da bomo pospešili razvoj aktivnega portala slovenske javne uprave.

##### 3.1.2. Uporabniki in uporaba znanja

Uporabniki znanja, zapisanega v ontologiji ŽS, so lahko državljani ter poslovni subjekti, uradniki v javni upravi, zakonodajalci, razvijalci elektronskih rešitev, računalniških aplikacij in strokovnjaki področja oz. uredniki modela ŽS. Ontologija ŽS tako predstavlja osrednji repozitorij znanja (Slika 13).

Slika 13: Tok znanja med ontologijo življenjskih situacij ter različnimi uporabniki.



Vir: Povzeto po Cukjati et al., 2008.

Državljeni imajo svoje ŽS, ki jih želijo rešiti. Ker pa velikokrat ne vedo, kam se morajo obrniti, poiščejo osrednji portal uprave. Tam s pomočjo za njih prilagojenega iskalnika izvedo vse o reševanju njihove ŽS: katere storitve javne uprave bodo morali sprožiti, katere dokumente izpolniti, kam jih poslati, na koga se lahko obrnejo za podrobnejšo

razlago itd. Dodana vrednost modela bi bila, da bi bile pokrite vse možne okoliščine, v katerih se lahko znajde državljan.

*Uradniki* v javni upravi rešujejo zadeve ali odgovarjajo na vprašanja uporabnikov, ki so lahko zelo različnih vrst. V vsakem primeru pa morajo poznati postopke, ki so lahko zelo zapleteni, z veliko posebnimi primeri. Tako imamo npr. v ŽS »Poročim se« možne situacije, ko je kakšna od oseb tujec, morda celo zunaj EU, mladoleten, sorodnik osebe, s katero se poroča, morda celo njegov posvojenec. Morda je bila oseba že prej poročena, morda v tujini, včasih je priča tujec, morda je katera od oseb gluha, nema ali slepa. Vsaka od teh okoliščin bolj ali manj spremeni reševanje ŽS oz. povzroči, da je treba priložiti dodatne dokumente. Z implementiranim modelom ŽS ter primernim iskalnikom bi imel uradnik ves čas na voljo vsa znanja o ŽS, storitvah javne uprave in dokumentih, ki jih potrebuje.

*Zakonodajalci* imajo nalogo, da sprejemajo *zakonodajo*, ki ureja in varuje življenje državljanov. Pri tem je pomembno, da so postopki javne uprave, ki so potrebni za izvajanje zakonov, učinkoviti, hitri in enostavni. Pri učinkovitem načrtovanju postopkov je nujno potrebno poznati obstoječe postopke, njihove slabosti ali pomanjkljivosti, ozka grla itd. Prav tako je koristno, če bi se dalo pri načrtovanju hitro simulirati predlagane rešitve, da bi se videlo prednosti ali slabosti novega zakona.

*Razvijalci elektronskih storitev* javne uprave pri svojem delu sami izdelajo natančen model ŽS oz. storitev, ki jih bo aplikacija izvajala. Znanje za model si pridobijo iz različnih virov, najpomembnejši pri tem pa so strokovnjaki področja. Model bi pomagal tako, da bi razvijalci lahko sami poiskali informacije, ki so že v implementiranem modelu oz. bi to naredili strokovnjaki. Izognili bi se večkratnemu modeliranju istih stvari.

*Aplikacije elektronskih storitev* javne uprave morajo poznati vse za izvajanje storitev potrebno znanje. Običajno je to znanje bolj ali manj togo zapisano v aplikaciji sami. Če pa bi se znanje o ŽS oz. storitvah javne uprave v fazi razvoja aplikacije zapisalo v implementiran model ŽS, bi lahko aplikacija to znanje črpala iz nje med izvajanjem. Znanja torej ne bi bilo treba zapisovati v aplikacijo. Ob morebitni spremembi zakonodaje bi uredniško popravilo obsegalo samo implementacijo modela ŽS, ne bi pa bilo potrebno programersko posegati v aplikacijo.

*Eksperti za področje javne uprave* so tisti, ki vedo, kako potekajo postopki, kdo je za kaj odgovoren, kateri zakon je podlaga za kakšen postopek, kako se postopki spremenijo, če se spremeni zakon, itd. Predstavljajo torej vir informacij za zakonodajalce, ki na osnovi teh informacij spreminjajo, odpravljajo ali sprejemajo nove zakone, po drugi strani so vir informacij za uradnike oz. vodje upravnih oddelkov in so vir informacij za razvijalce elektronskih rešitev, ker poznajo vsebino, ki mora biti informatizirana. Model ŽS bi vse to znanje strokovnjakov s področja uprave zbral na enem mestu ter omogočil dostop do njega zakonodajalcem, uradnikom, razvijalcem elektronskih rešitev ali pa bi ga uporabljali strokovnjaki, ko bi pomagali prej naštetim strankam.

### 3.2. ANALIZA OKOLJA

Namen analize okolja je identificirati tehnološko okolje, kjer se bo uporabljala ontologija ŽS.

Podatke o tehnološkem okolju smo dobili v poročilu projekta OneStopGov<sup>46</sup>, katerega cilj je razviti platformo za javno upravo, ki bi prek spleta omogočala vse na enem mestu (one-stop). V okviru tega projekta je bil izpolnjen vprašalnik o infrastrukturi, predvsem za potrebe portala javne uprave, podatki pa so povzeti po dokumentu OneStopGov: D21 (2006). Glede na to, da bi bil portal javne uprave glavni kanal za uporabo znanja, zapisanega v ontologiji ŽS, bi bila tudi sama ontologija verjetno postavljena v podobno, če že ne isto okolje. Zato lahko ugotovitve glede tehnološkega okolja portala vzamemo tudi za okolje ontologije.

Tehnološko okolje, v katerem bo ontologija nameščena, je torej naslednje:

- Inštitucije javne uprave so povezane z varnim omrežjem, imenovanim HKOM (Hitro Komunikacijsko Omrežje).
- Znotraj tega omrežja so javno dostopne le storitve, ki so locirane v posebni coni DMZ (demilitarized zone).
- Veliko inštitucij gostuje svoje aplikacije na infrastrukturi MJU.
- Večina e-storitev HKOM za državljane in gospodarske entitete je implementiranih kot samostojne aplikacije s svojimi vmesniki.
- Zaledne (angl. back-office) storitve so ravno tako implementirane na različne načine z lastnimi infrastrukturami z različnimi tehnologijami.
- Večina podatkov, ki bi jih potrebovali pri izvajanju ŽS, je shranjenih v relacijskih bazah podatkov: Oracle relational databases in Oracle Internet Application Server.
- Večina aplikacij e-uprave temelji na J2EE (Java 2 Platform, Enterprise Edition).
- Komunikacije med e-storitvami so urejane z orodji JDBC (Java Database Connectivity) ali potekajo po protokolu HTTPS, včasih pa kar po navadni e-pošti.
- Format e-dokumentov je lahko .doc, .pdf, .xml, .rtf in .tiff.
- Protokola za prenos podatkov sta HTTP in SOAP (Service-Oriented Architecture and Programming).
- Varnostni mehanizmi vključujejo: SSL, uporabniške certifikate ter digitalni podpis.
- Operacijski sistem je Unix, Linux ali Sun SPARC Solaris.

### 3.3. ANALIZA IZVEDLJIVOSTI

Namen analize izvedljivost je odgovoriti na vprašanje, ali je izgradnja ontologije, kot jo želimo, sploh izvedljiva. Ker je v tem magistrskem delu predstavljen prototip in ne polno zmogljiva ontologija, bom ponekod naredil analizo za obe možnosti.

Pri izvedljivosti imamo štiri vidike, na katere moramo biti pozorni<sup>47</sup>: operativna, tehnična, časovna in ekonomska izvedljivost.

---

<sup>46</sup> <http://www.onestopgov-project.org/index.php>

<sup>47</sup> <http://www.aillab.si/blaz/predavanja/pois/slides/pois10-AnalizaStroskovInKoristi.ppt>

### 3.3.1. Operativna izvedljivost

Operativna izvedljivost meri nujnost reševanja problema in sprejemljivost rešitve oz. kako dobro bo novi sistem rešil probleme v organizaciji. Pri tem imamo tri poglede (Tabela 5):

Tabela 5: Operativna izvedljivost.

|                                                            | Prototip ontologije ŽS, kot je načrtovan v tem magistrskem delu | Polno zmogljiva ontologija ŽS |
|------------------------------------------------------------|-----------------------------------------------------------------|-------------------------------|
| Je problem dovolj pereč, da ga moramo rešiti?              | DA                                                              | DA                            |
| Bo naš sistem ta problem rešil?                            | NE                                                              | DA                            |
| Kako uporabnik gleda (čuti) na problem in njegovo rešitev? | ?                                                               | ?                             |
| Analiza uporabnosti uporabniškega vmesnika                 | ?                                                               | ?                             |
| SKUPAJ                                                     | NE                                                              | DA                            |

Ugotovitev: prototip sam ne zagotavlja operativne izvedljivosti, polno zmogljiva ontologija pa bi jo zagotavljala. Ker je namen prototipa le pokazati pot do polno zmogljive ontologije, ki je šele končni cilj, je relevantno gledati ugotovitve za polno zmogljivo ontologijo.

### 3.3.2. Tehnična izvedljivost

Tehnična izvedljivost zajema praktično izvedljivost projekta, dostopnost tehničnih virov in kadrov (Tabela 6):

Tabela 6: Tehnična izvedljivost.

|                                                      | Prototip ontologije ŽS, kot je načrtovan v tem magistrskem delu                                  | Polno zmogljiva ontologija ŽS                                                                    |
|------------------------------------------------------|--------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|
| Ali sta predlagana rešitev in tehnologija praktični? | DA, ker temeljita na najširše sprejetih standardih oz. tehnologijah (OWL za predstavitev znanja) | DA, ker temeljita na najširše sprejetih standardih oz. tehnologijah (OWL za predstavitev znanja) |
| Ali trenutno razpolagamo s potrebno tehnologijo?     | DA                                                                                               | DA                                                                                               |
| Ali razpolagamo s potrebnimi kadri?                  | DA                                                                                               | DA, če bi se projekta lotili na MJU-ju                                                           |
| Ali je časovni raspored primeren?                    | DA                                                                                               | ? (ker še ni projekta)                                                                           |
| SKUPAJ                                               | DA                                                                                               | DA                                                                                               |

Ugotovitev: tako prototip kot polno zmogljiva ontologija sta tehnično izvedljivi.

### 3.3.3. Časovna izvedljivost

Časovna izvedljivost podaja oceno, ali je časovni načrt projekta primeren:

Tabela 7: Časovna izvedljivost.

|                                      | Prototip ontologije ŽS, kot je načrtovan v tem magistrskem delu | Polno zmogljiva ontologija ŽS |
|--------------------------------------|-----------------------------------------------------------------|-------------------------------|
| Čas, potreben za razvoj?             | 2 meseca                                                        | 6 mesecev                     |
| Ali je dolžina razvoja sprejemljiva? | DA                                                              | DA                            |
| SKUPAJ                               | DA                                                              | DA                            |

Ugotovitev: časovni načrt prototipa kot polne ontologije je izvedljiv in sprejemljiv.

### 3.3.4. Ekonomska izvedljivost

Ekonomska izvedljivost ali analiza stroškov in koristi (angl. cost-benefit analysis) podaja oceno, kako koristen oziroma praktičen bo informacijski sistem za organizacijo naročnika z ekonomskega vidika. Je ključnega pomena pri večini projektov. V začetnih fazah projekta, kjer v tem trenutku tudi smo, se analiza ekonomske izvedljivosti opira bolj na ugibanja, ali bodo rešitve, ki jih bo prinesel projekt, tudi dejansko uporabne.

Stroški se delijo na stroške razvoja in stroške obratovanja. Stroški obratovanja se nadalje delijo na stalne stroške, katerih višina je neodvisna od količine uporabe razvitega sistema, ter variabilne stroške, katerih višina je sorazmerna z obsegom uporabe (Tabela 8):

*Tabela 8: Stroški razvoja ter obratovanja.*

| <b>Stroški razvoja</b>              |                 | <b>(EUR)</b>                                  |
|-------------------------------------|-----------------|-----------------------------------------------|
| Ontologist (3 meseci, 2 osebi)      | 2.000 EUR/mesec | $2 \times 3 \times 2.000 = 12.000$            |
| Strok. javne uprave (3 mes., 2 os.) | 2.000 EUR/mesec | $2 \times 3 \times 2.000 = 12.000$            |
| Protege strok. (1 mesec)            | 3.000 EUR/mesec | $1 \times 1 \times 3.000 = 3.000$             |
| Sist. admin. (1/4 meseca)           | 2.000 EUR/mesec | $\frac{1}{4} \times 1 \times 2000 = 500$      |
| Programer (1 mesec)                 | 2.000 EUR/mesec | $1 \times 1 \times 2000 = 2000$               |
| Strok. za baze pod. (1 mes.)        | 2.000 EUR/mesec | $1 \times 1 \times 2000 = 2000$               |
| Server + software                   | 5.000 EUR       | 5.000                                         |
| Baza podatkov + client              | 5.000 EUR       | 5.000                                         |
| <b>SKUPAJ stroški</b>               |                 | <b>41.500</b>                                 |
|                                     |                 |                                               |
| <b>Letni obratovalni stroški</b>    |                 |                                               |
| Ontologist (polovično)              | 2.000 EUR/mesec | $\frac{1}{2} \times 12 \times 2.000 = 12.000$ |
| Strok. jav. upr. (polovično)        | 2.000 EUR/mesec | $\frac{1}{2} \times 12 \times 2.000 = 12.000$ |
| Licenca server                      | 500 EUR         | 500                                           |
| Licenca baza podatkov               | 500 EUR         | 500                                           |
| Potni stroški, pisarna ...          | 1.000 EUR/mesec | 12.000                                        |
| <b>SKUPAJ stroški</b>               |                 | <b>37.000</b>                                 |

Pri oceni količine potrebnega dela sem upošteval, da je področje močno formalizirano zaradi urejanja prek zakonodaje ter da je obenem dokaj nespremenljivo, ker je praktično za vsako večjo spremembo potreben politični konsenz. Prvo dejstvo olajšuje aktivnost konceptualizacije in formalizacije, drugo pa aktivnost zbiranja znanja, še posebej v času po implementaciji.

Koristi, ki bi jih zagotavljal delujoč sistem, lahko delimo na posredne ter neposredne finančne koristi za naročnika, ki bi bil verjetno MJU (Ministrstvo za javno upravo) ter na posredne in neposredne finančne koristi za končne uporabnike – državljane in poslovne subjekte (Tabela 9).

Tabela 9: Posredne in neposredne koristi za naročnika ter uporabnike.

| <b>Neposredne finančne koristi za naročnika (MJU) - letno</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | <b>(EUR)</b>                                                                       |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|
| Ker bi se vsaka informacija zapisala samo enkrat, ker ontologija to eksplicitno zahteva oz. preverja redundanco informacij, bi se olajšale posodobitve baze znanja (ko se npr. spremeni zakonodaja) in jo potem uporablja portal e-uprava. Prihranek na strošku zaposlenega.                                                                                                                                                                                                                                                                                                                                                                                                      | $12 \times 2.000 = 24.000$                                                         |
| Zaradi izčrpnije baze znanja bi bilo po e-pošti postavljenih manj vprašanj s strani državljanov, kar bi pomenilo prihranek v plači zaposlenega.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | $12 \times 2.000 = 24.000$                                                         |
| Pri razvoju novih e-storitev bi se razvoj pocenil zaradi lažjega modeliranja postopkov, ki bi bili vsaj delno že zmodelirani. Zelo groba ocena teh prihrankov je ena plača zaposlenega.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | $12 \times 2.000 = 24.000$                                                         |
| V primeru, da bi bila ontologija ŽS zastavljena tako, da bi omogočala tudi semantično povezljivost e-storitev, bi se razvoj e-storitev zelo pocenil, ker bi ontologija prek zagotavljanja semantične povezljivosti e-storitev podprla pristop SOA (service-oriented-architecture) ter s tem omogočila modularni pristop k informatizaciji storitev. Zelo groba ocena teh prihrankov je ena plača zaposlenega (trenutno nimam boljše ocene).                                                                                                                                                                                                                                       | $12 \times 2.000 = 24.000$                                                         |
| <b>Posredne koristi za naročnika (MJU)</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | <b>(EUR)</b>                                                                       |
| Na enem mestu zbrano vse znanje o storitvah in življenjskih okoliščinah poveča preglednost ter preprečuje podvajanje tega znanja. Poleg tega povečuje zanesljivost znanja, ker ga bi uporabljalo ter preverjalo več uporabnikov.                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Večja preglednost in zanesljivost znanja o storitvah in ŽS                         |
| Z informacijami bi bile pokrite vse možne okoliščine, v katerih se lahko znajde državljan, kar bi povečalo verodostojnost portala e-uprava.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Večja verodostojnost portala e-uprava                                              |
| Uradniki bi lažje in hitreje prišli do informacij.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Večja in lažja dostopnost informacij za uradnike                                   |
| Specialisti za javno upravo bi dobili orodje, s katerim bi si pomagali pri analizi postopkov. Formalni model bi namreč preprečeval različne interpretacije zakonodaje, če pa bi različne interpretacije vseeno obstajale, bi na ta način identificirali dvoumno zakonodajo. Imeli bi tudi konkreten in zanesljiv model, s pomočjo katerega bi lažje optimizirali postopke ter odpravljali administrativne ovire.                                                                                                                                                                                                                                                                  | Specialisti za javno upravo bi dobili praktično orodje za analizo postopkov in ŽS. |
| <b>Neposredne finančne koristi za uporabnike</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | <b>(EUR)</b>                                                                       |
| Prihranek časa za dostop do informacij zaradi popolnejše zbirke informacij na portalu e-uprava. Če na pamet ocenimo, da vsak dostop do portala e-uprava pomeni minuto prihranka (verjetno je pravilnejša ocena višja), in če upoštevamo, da ima portal e-uprava mesečno 100.000 obiskov <sup>48</sup> , potem lahko ocenimo, da je skupni prihranek za uporabnike 1666 ur. Če predpostavimo, da bi se uporaba portala zaradi ontologije povečala za 30 %, potem je prihranek uporabnikov okoli tri mesečne delovne obveznosti zaposlenega. (verjetno bi bilo bolj smiselno predpostavljati večje povečanje uporabe). Če gre za zaposlene, je prihranek za gospodarstvo tri plače. | $12 \times 3 \times 2000 = 72.000$                                                 |
| Prihranek uporabnikov zaradi informatiziranosti storitev javne uprave. Predpostavimo, da bi ontologija pospešila razvoj novih e-storitev (zaradi omogočanja semantične povezljivosti e-storitev) za 1 %. Če upoštevamo še, da je letni prihranek za uporabnike zaradi vzpostavitve informacijskega sistema e-VEM in sprememb povezane zakonodaje ocenjen na 10 milijonov letno (Kodra in Patekar, 2008) <sup>49</sup> , bi izgradnja ontologije pomenila prihranek 100.000 EUR letno.                                                                                                                                                                                             | $1 \% \text{ od } 10.000.000 = 100.000$                                            |
| <b>Posredne koristi za ostale uporabnike</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | <b>(EUR)</b>                                                                       |
| Državljeni – imeli bi večji občutek gotovosti, ker bi lahko na enem mestu kadarkoli dobili informacije o postopkih in ŽS, pri čemer bi bile z informacijami pokrite vse možne okoliščine, v katerih se lahko znajde državljan.                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Večja enostavnost dostopa do informacij – večje zadovoljstvo.                      |

<sup>48</sup> Ta podatek je razviden iz analize dostopov do portala e-uprava, ki so mi ga posredovali iz MJU.

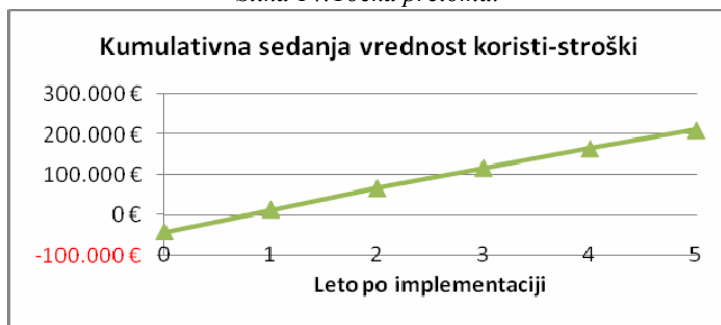
<sup>49</sup> Gre za prihranke iz naslova poenostavljenega postopka ustanovitve, sprememb ali izbrisa enostavne družbe z omejeno odgovornostjo, prijave, odjave ali sprememb iz naslova obveznih socialnih zavarovanj ali podatkov o družinskih članih, poenostavljenega postopka prijave delovnega mesta oziroma vrste dela in vzpostavitev javne knjige, v skupni višini 10,24 milijona EUR na leto glede na stanje pred uvedbo informacijskega sistema e-VEM.

Sedaj lahko naredimo tabelo neposrednih stroškov in koristi za naročnika po letih (Tabela 10), s čimer bomo prišli do kumulativne vrednosti projekta po letih, ki bo pokazala točko preloma (Slika 14).

Tabela 10: Neposredni stroški in koristi za naročnika po letih.

| Leto                                           | 0         | 1         | 2          | 3          | 4          | 5          |
|------------------------------------------------|-----------|-----------|------------|------------|------------|------------|
| <b>Stroški</b>                                 |           |           |            |            |            |            |
| Stroški razvoja                                | -41.500 € | 0 €       | 0 €        | 0 €        | 0 €        | 0 €        |
| Obratovalni stroški (inflacija 5 % upoštevana) | 0 €       | -37.000 € | -38.850 €  | -40.793 €  | -42.832 €  | -44.974 €  |
| Obrestni faktor (10% letno)                    | 1         | 1,1       | 1,21       | 1,331      | 1,4641     | 1,61051    |
| Sedanja vrednost stroškov                      | -41.500 € | -33.636 € | -32.107 €  | -30.648 €  | -29.255 €  | -27.925 €  |
| Kumulativna sedanja vrednost stroškov          | -41.500 € | -75.136 € | -107.244 € | -137.892 € | -167.147 € | -195.072 € |
| <b>Koristi</b>                                 |           |           |            |            |            |            |
| Koristi (inflacija 5 % upoštevana)             | 0 €       | 98.000 €  | 102.900 €  | 108.045 €  | 113.447 €  | 119.120 €  |
| Sedanja vrednost koristi                       | 0 €       | 89.091 €  | 85.041 €   | 81.176 €   | 77.486 €   | 73.964 €   |
| Kumulativna sedanja vrednost koristi           | 0 €       | 89.091 €  | 174.132 €  | 255.308 €  | 332.794 €  | 406.758 €  |
| <b>Skupaj (koristi-stroški)</b>                |           |           |            |            |            |            |
| Kumulativna sedanja vrednost koristi-stroški   | -41.500 € | 13.955 €  | 66.888 €   | 117.416 €  | 165.647 €  | 211.686 €  |

Slika 14: Točka preloma.



Točka preloma je eno leto po implementaciji.

Neto sedanja vrednost po petih letih je okoli 200.000 EUR.

Investicijski donos v prvih petih letih je

$$Investicijski\_donos = \left( \frac{407.000 - 195.000}{195.000} \right) = 108\%$$

Preden podamo končno ugotovitev, bi rad opozoril še na dve dejstvi. Prvo je, da je več postavk v izračunu zelo grobih in temu primerne so grobe tudi ocene točke preloma, neto sedanje vrednosti in investicijskega donosa. Drugo dejstvo pa je, da namen javne uprave ni le ustvarjanje nove vrednosti oz. dobičkonosnost, temveč tudi ali predvsem servisiranje državljanov in urejanje dogajanja v državi. Zato se pri končnem zaključku v aktivnosti analize stroškov/koristi ne bi smelo upoštevati le neposrednih finančnih koristi naročnika, temveč tudi vse ostale koristi uporabnikov.

Ugotovitev: projekt polno zmogljive ontologije ŽS je ekonomsko opravičljiv. Upoštevanje posrednih in neposrednih koristi uporabnikov to ugotovitev še dodatno podkrepi.



### 3.4. ZBIRANJE ZNANJA

V aktivnosti *zbiranje znanja* bomo poskusili pridobiti čimveč oz. dovolj znanja o področju proučevanja. V našem primeru so to storitve javne uprave in ŽS državljanov ali poslovnih subjektov, katerih reševanje zahteva izvajanje storitev javne uprave.

Možni viri znanja, ki ga želimo modelirati, so zakonodaja, ki ureja področje, uradniki, ki dnevno rešujejo zadeve v določenih postopkih, strokovno-znanstveni članki na temo modeliranja ŽS, že razviti in delujoči portali, npr. portal e-uUprava, projekti s področja razvoja naprednejših upravnih portalov ... Še posebej so koristni viri, ki opisujejo modele ali sheme javne uprave, ker običajno že vsebujejo določeno mero konceptualizacije oz. urejajo in izpostavljajo za področje proučevanja pomembne pojme. Po drugi strani so vse informacije, potrebne za izvajanje postopkov javne uprave, načeloma zapisane v zakonodaji, vendar njeno proučevanje presega usposobljenost ontologistov. Zato so uradniki ali strokovnjaki javne uprave neprecenljiv vir informacij, ko gre za konkretne primere postopkov in njihovih variant. Pri podrobni analizi postopkov brez njihove pomoči ontologist težko pride do relevantnih informacij. Zato smo najprej proučili projekte s področja razvoja naprednejših upravnih portalov, nato že delujoči portal e-uprava. Intervjuvanje strokovnjakov in uradnikov pa smo si prihranili za situacije, ko na kakšen drug način nikakor ne bi prišli do želenih informacij.

#### 3.4.1. Raziskovalni projekti

Veliko informacij oz. znanja o področju ŽS smo našli v dokumentih projekta OneStopGov<sup>50</sup>, v okviru katerega je bila tudi razvita metodologija za zajem znanja s področja ŽS (Todorovski et al., 2006), ki je potrebno za implementacijo aktivnega portala ŽS ob upoštevanju vseh možnih okoliščin državljana in je bila podrobneje predstavljena v poglavju 1.3.5.

Metodologija predlaga analizo znanja o ŽS s pomočjo treh tabel, v katere se vpisujejo informacije o ŽS, postopkih in dokumentih. V projektu je bilo tako analiziranih 16 ŽS, po štiri v štirih različnih državah, med katerimi je bila tudi Slovenija. V poročilu D12<sup>51</sup> (Todorovski, 2007) so podane izpolnjene tabele, ki jih je za okoli 140 strani formata A4.

Za potrebe tega dela je po eni strani zanimivo znanje na konceptualnem nivoju (o ŽS, postopkih, dokumentih in njihovih atributih) ter po drugi strani znanje o konkretnih ŽS, kot so »Poročim se«, »Organiziranje javne prireditve«, »Ustanovitev s. p.« ter »Ustanovitev podjetja za katering«.

V projektu OneStopGov lahko najdemo še eno vrsto znanja: znanje o poteku storitev javne uprave znotraj določene ŽS. Pri tem gre predvsem za informacije o zaporedju postopkov in pogojih za njihovo sprožanje. Primer grafičnega modela ŽS »Poročim se« je predstavljen na Slika 15. Pomembno je še dejstvo, da so zapisane tudi vse povezave med potrebnimi dokumenti in postopki, v katerih ti dokumenti nastanejo kot glavni izhodi (Slika 16 in

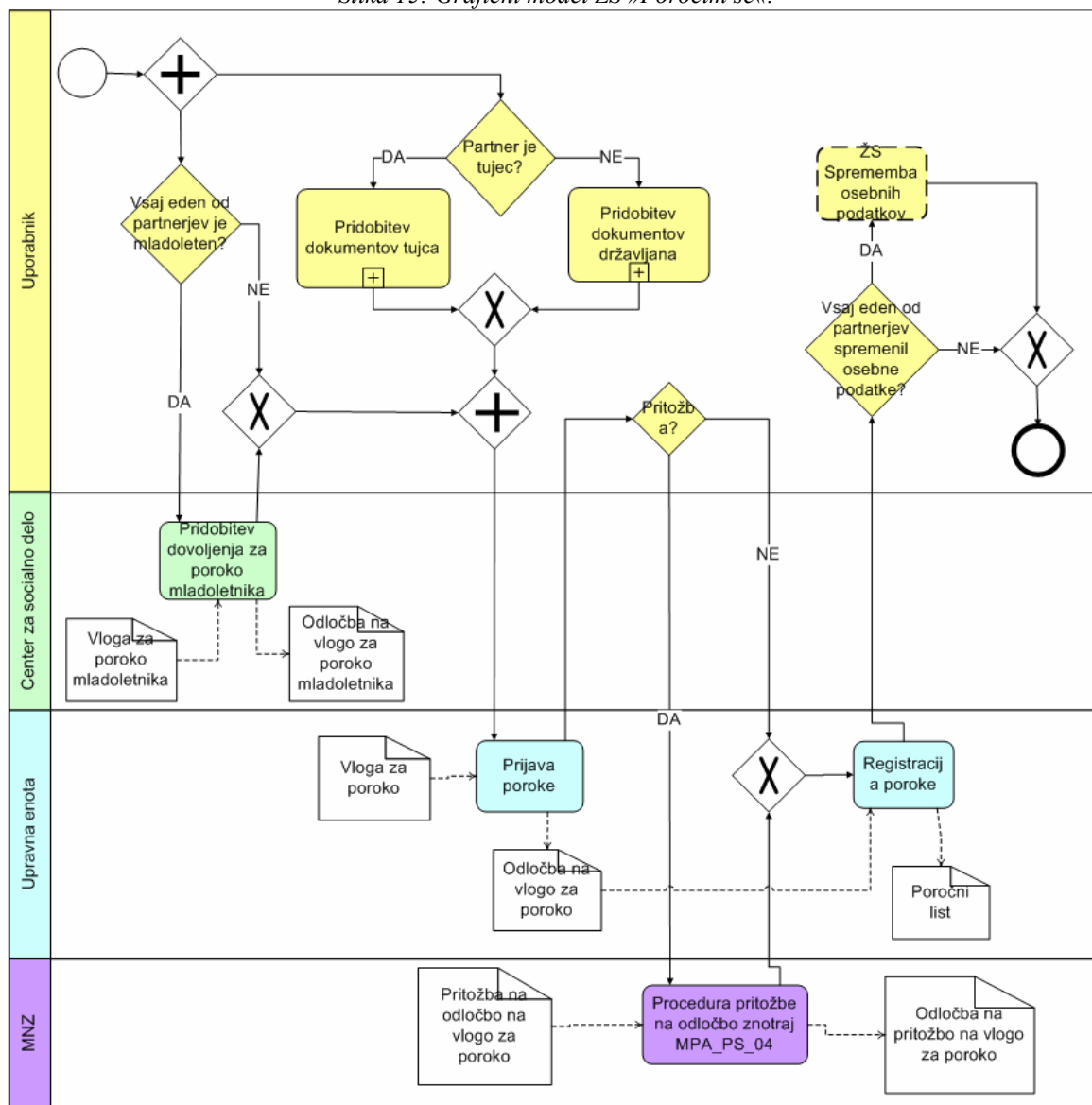
---

<sup>50</sup> OneStopGov – A life-event oriented framework and platform for one-stop government. IST STREP PROJECT OneStopGov, FP6-2004-IST-4-026965. Link: <http://www.onestopgov-project.org/index.php>

<sup>51</sup> <http://www.onestopgov-project.org/index.php?name=UpDownload&req=getit&lid=459>

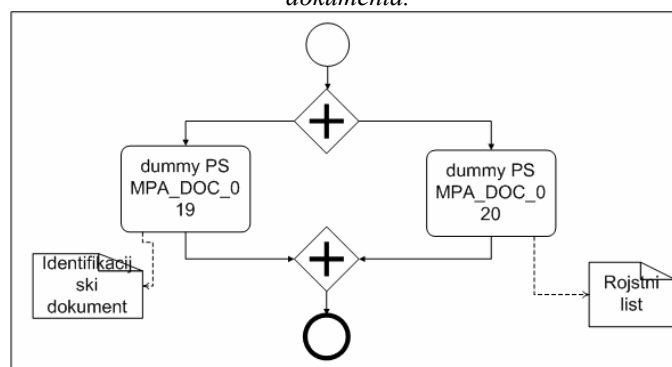
Slika 17).

Slika 15: Grafični model ŽS »Poročim se«.



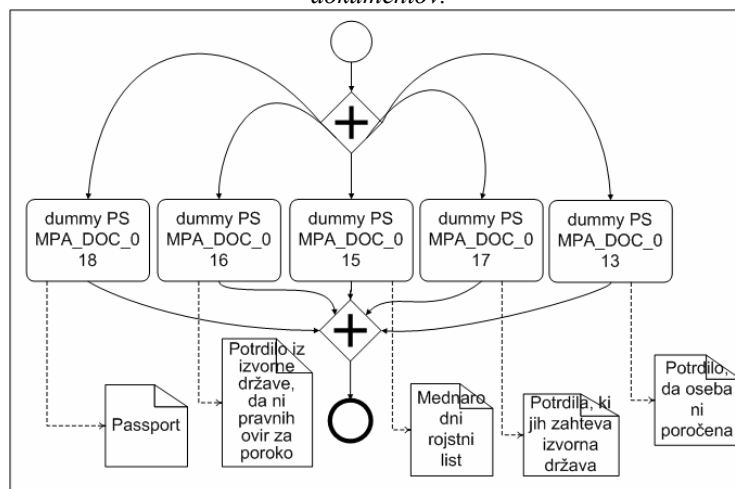
Vir: Todorovski, 2007.

Slika 16: Grafični model navidezne storitve »Pridobitev dokumentov državljana«, katere rezultat sta dva dokumenta.



Vir: Todorovski, 2007.

Slika 17: Grafični model navidezne storitve »Pridobitev dokumentov tujca«, katere rezultat je več dokumentov.



Vir: Todorovski, 2007.

Naš namen je, da bi v ontologijo, ki jo razvijamo, vnesli tudi tovrstne informacije, s čimer bi omogočili avtomatično vodenje uporabnika pri reševanju njegove ŽS ob upoštevanju vseh njegovih specifičnih okoliščin. Vsi dosednji poskusi (po našem poznavanju) so se namreč končali z izdelavo množice podprimerov, kjer je vsak podprimer posebej zakodiran v aplikaciji.

### 3.4.2. Delujoči upravni portali

Pri iskanju informacij in znanja o ŽS, njim pridajajočih postopkih javne uprave ter dokumentih smo proučili tudi obstoječe portale javne uprave, predvsem slovenski portal e-uprava<sup>52</sup>, ki je podrobneje predstavljen v poglavju 1.2.4.

Vsak tak portal gotovo vsebuje neko obliko organiziranja ali združevanja postopkov, običajno v ŽS, ki so nadalje urejene po skupinah, kot je to v primeru portala e-uprava (Slika 2). Uporabne informacije za potrebe tega dela so še: atributi ŽS, povezave na pravne podlage, nekaj variant na najvišjem nivoju, ponekod povezava na vloge ali celo na e-storitev, če je že razvita.

Na portalu e-uprava je obdelanih več ŽS kot v projektu OneStopGov, vendar manj podrobno, ker poudarek ni na pokrivanju vseh možnih okoliščin, v katerih je lahko uporabnik, temveč na pokrivanju vseh ŽS. Tako nimamo v primeru poroke mladoletne osebe na portalu skoraj nobenih informacij, kaj, če sploh kaj, je treba storiti, preden se taka oseba lahko formalno poroči (Slika 18).

<sup>52</sup> <http://e-uprava.gov.si/e-uprava/>

Slika 18: Opis življenjske situacije »Poročim se«.

|                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-----------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <b>NAZIV STORITVE</b><br><b>Sklenitev zakonske zveze slovenskih državljanov v RS</b><br>(Zadnja sprememba: 23.05.2007)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|  | <b>ORGAN, KI JO UREJA</b><br>MINISTRSTVO ZA DELO, DRUŽINO IN SOCIALNE ZADEVE                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|  | <b>UVOD</b><br>Zakonska zveza je z zakonom urejena življenjska skupnost moža in žene. Temelji na svobodni odločitvi skleniti zakonsko zvezo, na obojestranski čustveni navezanosti, vzajemnem spoštovanju, razumevanju, zaupanju in medsebojni pomoči. V zakonski zvezi sta zakonca enakopravna.                                                                                                                                                                                                                                                                                                               |
|  | <b>KDO</b><br>Dve <u>praviloma polnoletni osebi</u> različnega spola, ki se želita pročitati oziroma skleniti zakonsko zvezo.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|  | <b>KJE IN KAKO</b><br>Osebi, ki nameravata skleniti zakonsko zvezo, se prijavita matičarju, ki vodi poročno matično knjigo za naselje, kjer želita skleniti zakonsko zvezo. Osebi, ki nameravata skleniti zakonsko zvezo, v prijavi izjavita, da svobodno sklepata zakonsko zvezo in da so izpolnjeni pogoji za njeno veljavnost. Sklenitev zakonske zveze prijavitelja največ tri mesece in najmanj 14 dni pred datumom sklenitve zakonske zveze. Pri prijavi predložita izpiske iz rojstne knjige in potrdilo oz. obvestilo o državljanstvu, če teh podatkov ni mogoče ugotoviti na podlagi uradnih evidenc. |
|  | <b>PRISTOJNI ORGANI</b><br>Za politiko in razvoj:<br>Ministrstvo za delo, družino in socialne zadeve                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

Vir: portal e-uprava, nov. 2008.

### 3.4.3. Ostala znanstvena literatura

Pregledali smo tudi strokovno-znanstveno literaturo na temo semantičnega povezovanja e-storitev, ki je podrobneje predstavljena v poglavju 1.3.2. Ta vir znanja je zanimiv predvsem zaradi idej o možnih primerih uporabe ter povezavi različnih tehnologij, kot so SWS (Semantic Web Services), XML-modeli e-storitev, uporaba jezikov za zapis težkokategornih ontologij (npr. OWL). Pomanjkljivost tega vira znanja pa je, da se večinoma osredotoča na e-storitve in ŽS niti ne omenja, razen v nekaterih primerih (glej Tabela 1).

### 3.4.4. Zakonodaja

V primeru nejasnosti smo pogledali zakonodajo, ki ureja proučevano področje. Čeprav je jasnost zakonodajnih besedil včasih vprašljiva (Cukjati in Rakar, 2009), lahko predpostavimo, da je načeloma vse znanje o pravicah, obveznostih in postopkih, ki ga potrebujemo za ureditev kakršnekoli upravne zadeve, zapisano v uradno sprejetih zakonskih in podzakonskih aktih.

Primer takega znanja je informacija o tem, kako poteka postopek, kadar se poroča mladoletna oseba. Ta informacija je v 24. čl. Zakona o zakonski zvezi in družinskih razmerjih (ZZZDR-UPB1)<sup>53</sup> in pravi: »Preden center za socialno delo dovoli sklenitev zakonske zveze po 23. členu tega zakona mladoletniku, mora zaslišati njega, osebo, s katero namerava skleniti zakonsko zvezo, ter njegove starše oziroma skrbnika.«

Zakonodaja je izčrpen in zanesljiv vir znanja, ki pa ima kot vir znanja za potrebe tega dela več pomanjkljivosti. Nepoznavalcem področja pomeni branje zakonodaje časovno naporno opravilo, saj npr. samo prej omenjeni zakon (ZZZDR-UPB1) obsega 37 strani ali 12.000 besed ali 77.000 znakov. Poleg tega manjkajo vse informacije o ŽS, ker tega pojma v

<sup>53</sup> Zakon o zakonski zvezi in družinskih razmerjih, uradno prečiščeno besedilo (ZZZDR-UPB1) Uradni list RS, št. 69/2004 z dne 24. 06. 2004.

zakonodaji ni, ter vse informacije o konkretnem reševanju konkretne zadeve (npr. kje je upravna enota v Ljubljani).

#### **3.4.5. Uradniki**

Če se pri ekstrakciji znanja zatakne, vedno obstaja možnost, da o specifičnem področju ali vprašanju povprašamo tiste, ki se z določenim vprašanjem ukvarjajo vsak dan – uradnike, ki rešujejo določene zadeve, ker ti iz vsakodnevne prakse najboljše poznajo področje. Izkazalo se je, da je tudi za komunikacijo z njimi najprimerneje uporabljati metodologijo, predstavljeno v poglavju 1.3.5, saj le-ta zagotavlja, da se pogovor ne oddalji od informacij, ki jih v resnici potrebujemo pri gradnji ontologije. Pri pogovoru s strokovnjaki smo opazili še, da tudi oni relativno pogosto ne poznajo odgovorov na vsa vprašanja, ki se pri modeliranju pojavijo, so pa veliko hitrejši pri iskanju znanja, saj vedo, katera zakonodaja ureja določena vprašanja ali pa zelo hitro pridejo do kontaktnih podatkov drugih poznavalcev področja. Najtežje pri tem načinu zbiranja znanja je najti dovolj kompetentnega strokovnjaka, ki bi bil pripravljen porabiti dovolj časa in energije, da ontologist pridobi dovolj vpogleda v področje oz. znanja s področja proučevanja.

### **3.5. KONCEPTUALIZACIJA**

Konceptualizacija predstavlja aktivnost izdelovanja konceptualnega modela. V metodologiji METHONTOLOGY, po kateri razvijamo ontologijo ŽS, je v tej aktivnosti enajst korakov, ki so predstavljeni v tem pod poglavju.

#### **3.5.1. Prvi korak: Slovar izrazov**

Prvi korak je izdelava neformalnega slovarja izrazov, ki so pomembni za neko področje znanja. Treba jih je zapisati, jim določiti definicije ter sinonime in akronime.

Na podlagi vseh virov znanja, ki so omenjeni v poglavju 3.4, naredimo slovar izrazov na področju ŽS v javni upravi, ki se nam zdijo pomembni za namen ontologije, ki jo razvijamo. Pozorni moramo biti predvsem na izraze, ki se v različnih virih znanja že pojavljajo kot koncepti ali se vzporedno ponavljajo v več virih znanja, se zdijo strokovnjakom področja pomembni itd. Ti izrazi nam bodo v prihodnje pomenili koncepte, primerke, attribute konceptov in primerkov, relacije med njimi ... V tem koraku se še ne obremenjujemo s podvajanjem izrazov, če npr. dva različna izraza pomenita v bistvu en in isti koncept. Rezultat tega koraka je Tabela 11, kjer so izrazi navedeni neurejeno, kot so se pri proučevanju področja pojavljali, pri čemer so predstavljeni le pomembnejši izrazi:

*Tabela 11: Slovar izrazov za potrebe gradnje ontologije življenjskih situacij.*

| Ime                     | Sinonim             | Akronim  | Opis                                                              | Tip                  |
|-------------------------|---------------------|----------|-------------------------------------------------------------------|----------------------|
| Življenjska situacija   | Življenjski dogodek | ŽS, LE   | Situacija v življenju posameznika, v kateri se le-ta lahko znajde | koncept              |
| Življenjski dogodek     | Življenjski dogodek |          | Situacija v življenju posameznika, v kateri se le-ta lahko znajde | koncept              |
| Poslovna situacija      |                     |          | Situacija, v kateri se lahko znajde poslovni subjekt              | koncept              |
| Dokument                |                     |          | Dokument v papirni ali elektronski obliki                         | koncept              |
| Storitev javne uprave   |                     | SJU, PAS | Storitev, ki jo ponuja javna uprava                               | koncept              |
| SJU za državljane       |                     |          | SJU, ki jo sprožajo državljani, ko rešujejo svoje situacije       | koncept              |
| Postopek                | SJU                 |          | Storitev JU, gledano z zornega kota javne uprave                  | koncept              |
| Uporabnikova okoliščina |                     |          | Okoliščine državljan, ki vplivajo na potek reševanja ŽS           | koncept              |
| »Družina in otroci«     |                     |          | Skupina ŽS                                                        | koncept              |
| »Poročim se«            |                     |          | ŽS, ko se želita dva poročiti                                     | primerek             |
| Državljan               |                     |          | Državljan, ki rešuje ŽS                                           | koncept              |
| Podjetje                |                     |          | Podjetje, ki rešuje svojo poslovno situacijo                      | koncept              |
| Organ                   |                     |          | Organ javne uprave                                                | koncept              |
| Samostojni podjetnik    |                     |          | Ena od pravnih oseb                                               | koncept              |
| Organ javne uprave      |                     |          | Organ javne uprave                                                | koncept              |
| »Miha«                  |                     |          | Točno določen primerek državljan                                  | primerek             |
| Uslužbenec javne uprave |                     |          | Tisti, ki je zaposlen v javni upravi in deluje v njenem imenu     | koncept              |
| Zadeva                  |                     |          | Zadeva, ki jo rešuje organ JU                                     | koncept              |
| Št. zadeve              |                     |          | Številka zadeve                                                   | atribut koncepta     |
| Zakon                   |                     |          | Zakon, ki ureja neko področje                                     | koncept              |
| Povezana zakonodaja     |                     |          | Zakonodaja, ki določa reševanje ŽS ali SJU                        | lastnost koncepta    |
| Vloga                   |                     |          | Vsak postopek se sproži z vložitvijo vloge, ki je dokument        | koncept ali relacija |
| Priloga                 |                     | SD       | Dokument se v postopku lahko pojavi kot priloga določeni vlogi    | koncept ali relacija |
| Odločba                 |                     |          | Vsak postopek ima kot rezultat pozitivno ali negativno odločbo    | relacija             |
| Ponudnik storitve       |                     |          | Organ, ki ponuja ali rešuje določeno storitev                     | relacija             |
| Strošek                 |                     |          | Stroški za uporabnika pri reševanju zadeve                        | lastnost koncepta    |
| Izdajatelj              |                     |          | Tisti, ki izda dokument                                           | relacija             |
| Interesent              |                     |          | Tisti, ki je zainteresiran za posedovanje nekega dokumenta        | lastnost koncepta    |
| jePodpostopek           |                     |          | Nek postopek lahko nastopa kot podpostopek v postopku.            | relacija             |
| Pritožba                |                     |          | Dokument, s katerim se uporabnik pritoži na izdano odločbo        | koncept              |
| imaPritožbo             |                     |          | Povezuje določeno ŽS s storitvami javne uprave (SJU) ali ŽS       | relacija             |
| jeRezultat              |                     |          | Vsak dokument je rezultat nekega postopka                         | relacija             |

### **3.5.2. Drugi korak: Taksonomija konceptov**

V drugem koraku se iz izrazov v slovarju izrazov definirajo koncepti, ki se jih uredi v eni ali več taksonomijah.

Pri generalizaciji (superclass-of) ali specializaciji (subclass-of) razredov uporabimo štiri taksonomične relacije, kot jih predlaga metodologija METHONTOLOGY in ki so definirani v Ontologiji okvirjev (Frame ontology) in ontologiji OKBC: *Podrazred (angl.*

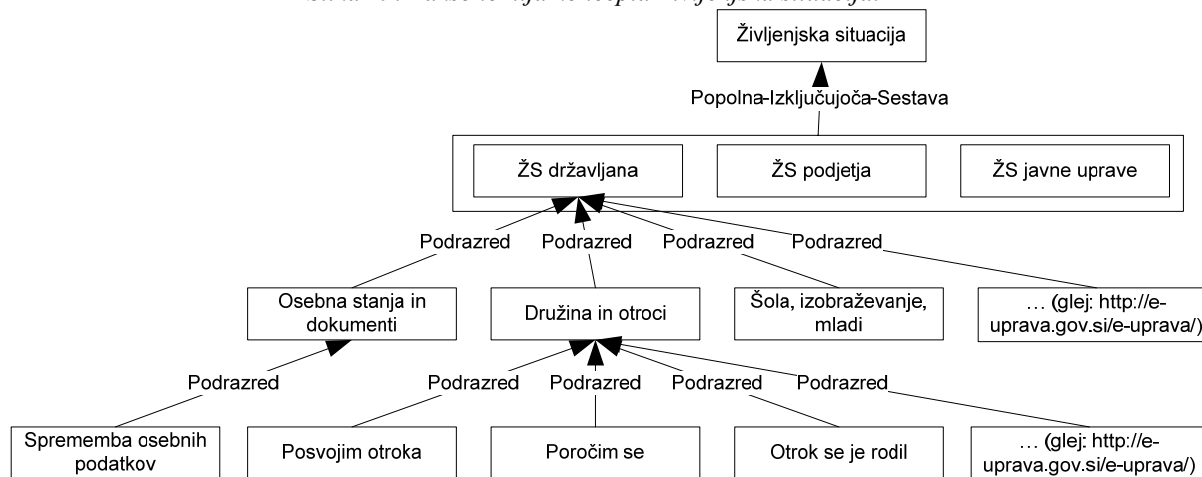
*Subclass-Of*), *Izključujoča-Sestava* (angl. *Disjoint-Decomposition*), *Popolna-Sestava* (angl. *Exhaustive-Decomposition*) in *Popolna-Izključujoča-Sestava* (angl. *Partition*). *Disjoint-Decomposition* označuje, da gre za izključujočo sestavo, kjer se naštetih podrazredov nekega nadrazreda med seboj izključujejo (vsak primerek nadrazreda lahko pripada le enemu podrazredu). *Exhaustive-Decomposition* označuje, da gre za popolno sestavo, kjer je vsak primerek nadrazreda primerek tudi vsaj enega od naštetih podrazredov. *Partition* nadrazreda v podrazrede pa je kombinacija izključujoče in popolne sestave, kar pomeni, da je vsak primerek nadrazreda primerek vsaj in točno enega podrazreda v particiji.

Glavni koncepti v ontologiji ŽS so:

- Življenjska situacija (krajše ŽS);
- Storitev;
- E-Storitev;
- Dokument;
- Uporabnik;
- Ponudnik storitve;
- Procedura<sup>54</sup>;
- Element procedure (krajše EP).

V nadaljevanju so predstavljene taksonomije glavnih konceptov, pri čemer koncepti E-Storitev, Procedura in Ponudnik storitev nimajo podrazredov. Prva dva zato, ker sta koncepta zelo enolična, tretji pa zato, ker bi bila taksonomija zelo razvejana<sup>55 56</sup>, pri čemer pa za samo ontologijo ŽS ni ključnega pomena. Po svojem bistvu bi morala biti taksonomija ponudnikov storitev del ontologije javne uprave, ko bo le-ta enkrat zgrajena.

Slika 19: Taksonomija koncepta Življenjska situacija.

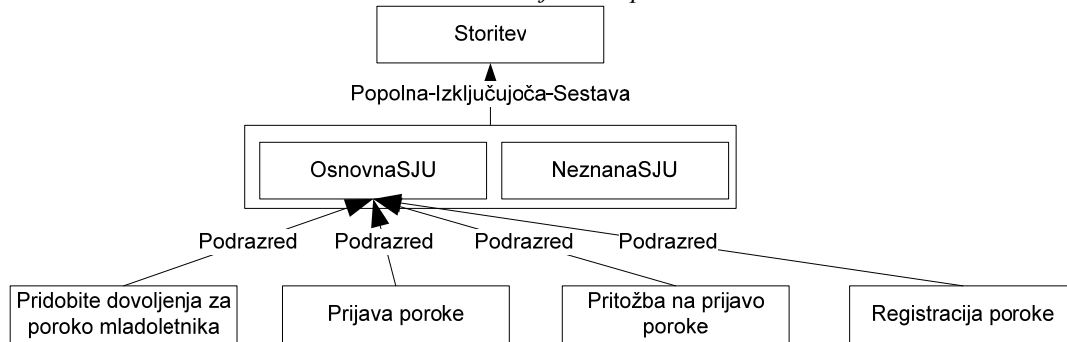


<sup>54</sup> Procedura in njeni elementi bi lahko bili zapisani v lastni ontologiji, ontologiji procedure, vendar jih bom zaenkrat vključil v ontologijo življenjskih situacij.

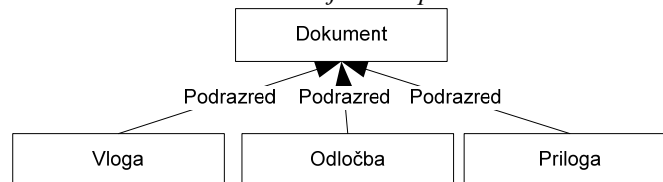
<sup>55</sup> <http://www.gov.si/abecedno.html>

<sup>56</sup> <http://e-uprava.gov.si/e-uprava/javni.euprava>

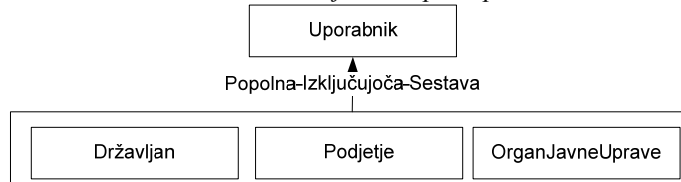
Slika 20: Taksonomija koncepta Storitev.



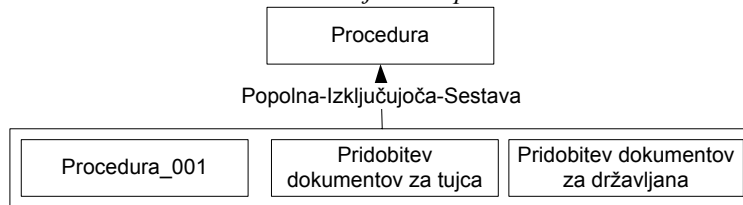
Slika 21: Taksonomija koncepta Dokument.



Slika 22: Taksonomija koncepta Uporabnik.

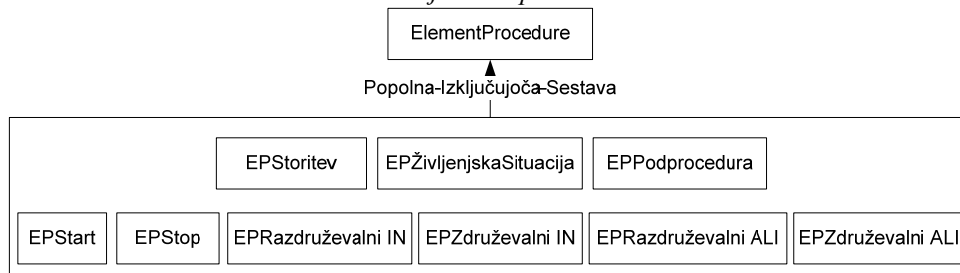


Slika 23: Taksonomija koncepta Procedura.



Največ težav imamo z reševanjem problema, kako zapisati znanje o procesu reševanja posamezne ŽS. Problem je torej v tem, kako zapisati znanje, ki je zajeto v diagramu poteka za posamezno ŽS, tako da se ne izgubi nobena za izvedbo postopka pomembna informacija. Tega se lotimo tako, da vse možne elemente v diagramu poteka pretvorimo v koncept `ElementProcedure` oz. njegove podrazrede (Slika 24).

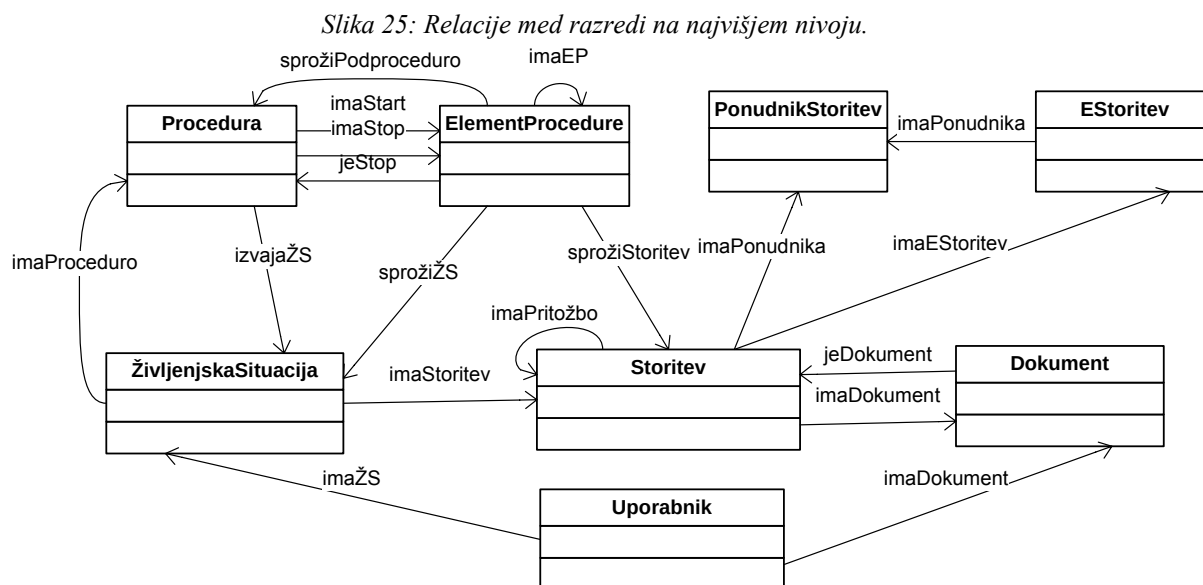
Slika 24: Taksonomija koncepta `ElementProcedure`.





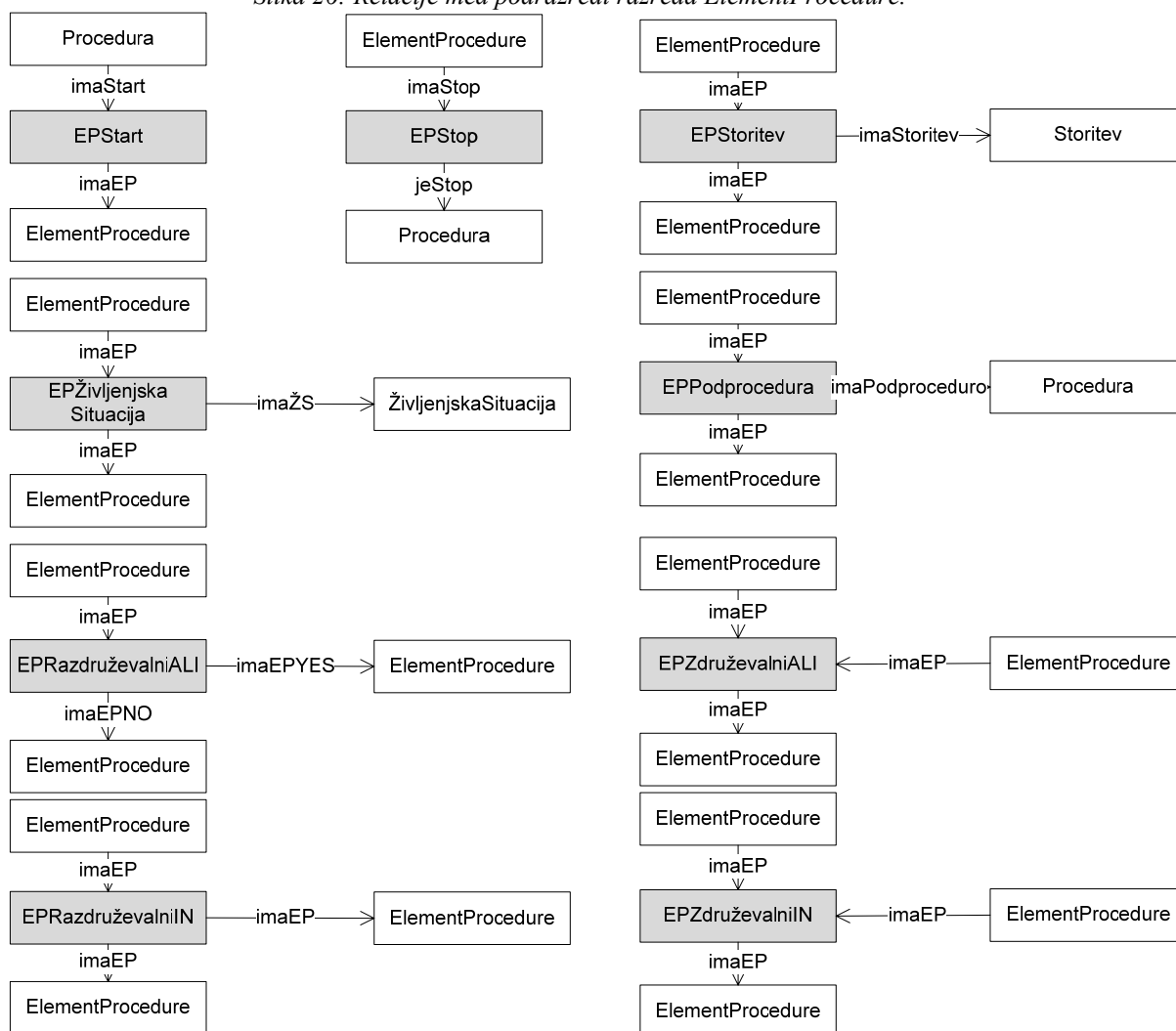
### 3.5.3. Tretji korak: Diagram ad-hoc binarnih relacij

V tretjem koraku se izdela neformalni diagram ad-hoc binarnih relacij med koncepti naše prihodnje ontologije in po potrebi tudi med koncepti naše ontologije ter koncepti drugih ontologij. Na spodnji sliki (Slika 25) so prikazani koncepti na vrhnjih nivojih in relacije, ki so pomembne pri reševanju ŽS.



Povezave, ki jih ontologija predvideva med posameznimi razredi koncepta ElementProcedure, si zaslužijo posebno pozornost. Kot je že bilo omenjeno, bi tovrstno znanje mirno lahko uvrstili v lastno ontologijo, ker gre za zaključeno celoto. Zaradi preglednosti so te relacije posebej razdelane na naslednji sliki (Slika 26), kjer so opazovani koncepti zaradi preglednosti obarvani s sivo barvo.

Slika 26: Relacije med podrazredi razreda *ElementProcedure*.



Koncepte naše ontologije bi lahko povezali s koncepti kakšne od vrhnjih ontologij, vendar smo se v tej fazi zaradi pomanjkanja časa odločili, da zaenkrat ne bomo povezovali konceptov ontologije, ki jo razvijamo, s koncepti drugih prosto dostopnih ontologij, kot je npr. SUMO (Standard Upper Merged Ontology).

### 3.5.4. Četrty korak: Slovar konceptov

V četrtem koraku se izdelata formalni slovar konceptov, ki vključuje prejšnje informacije (ime koncepta, njegova definicija, atributi in ad-hoc binarne relacije med njimi). Pri tem se predpostavlja, da imajo relacije in atributi lokalni doseg ter pripadajo le konceptu, ob katerem so navedeni. Zaradi tega se njihovo poimenovanje lahko ponavlja med različnimi koncepti. Rezultat tega koraka je tabela (Tabela 12), kjer so navedeni vsi pomembni koncepti naše ontologije:

Tabela 12: Slovar konceptov.

| Koncept                                                | Atributi razreda                                | Atributi primerka                                         | Relacije                                                                            |
|--------------------------------------------------------|-------------------------------------------------|-----------------------------------------------------------|-------------------------------------------------------------------------------------|
| ŽivljenjskaSituacija                                   | ime<br>opis<br>ciljnaSkupina<br>organZaUrejanje | ID                                                        | imaProceduro<br>imaStoritev                                                         |
| Storitev                                               | ime<br>pravnaPodlaga<br>takse<br>časIzvajanja   | ID                                                        | imaEStoritev<br>imaOdločbo<br>imaVlogo<br>imaPrilogo<br>imaPonudnika<br>imaPritožbo |
| EStoritev                                              | ime<br>hyperpovezava                            | ID                                                        | imaPonudnika                                                                        |
| PonudnikStoritve                                       | ime                                             |                                                           | -                                                                                   |
| Dokument                                               | ime                                             | ID<br>datumIzdaje                                         | jeRezultat                                                                          |
| Uporabnik                                              | -                                               | ID<br>imePriimek<br>datumRojstva<br>Stan<br>Državljanstvo | imaŽS<br>imaDokument                                                                |
| Procedura                                              | -                                               | ID<br>časStart<br>časStop                                 | imaStart<br>izvajaŽS<br>imaStop                                                     |
| ElementProcedure                                       | -                                               | ID<br>časStart<br>časStop                                 | -                                                                                   |
| EPStart<br>(podrazred ElementProcedure)                | -                                               | -                                                         | imaEP                                                                               |
| EPStop<br>(podrazred ElementProcedure)                 | -                                               | -                                                         | jeStop                                                                              |
| EPStoritev<br>(podrazred ElementProcedure)             | -                                               | -                                                         | sprožiStoritev<br>imaEP                                                             |
| EPŽivljenjskaSituacija<br>(podrazred ElementProcedure) | -                                               | -                                                         | sprožiŽS<br>imaEP                                                                   |
| EPPodprocedura<br>(podrazred ElementProcedure)         | -                                               | -                                                         | sprožiPodproceduro<br>imaEP                                                         |
| EPRazdruževalniIN<br>(podrazred ElementProcedure)      | -                                               | -                                                         | imaEP                                                                               |
| EPZdruževalniIN<br>(podrazred ElementProcedure)        | -                                               | -                                                         | imaEP                                                                               |
| EPRazdruževalniALI<br>(podrazred ElementProcedure)     | -                                               | -                                                         | imaEPYES<br>imaEPNO                                                                 |
| EPZdruževalniALI<br>(podrazred ElementProcedure)       | -                                               | -                                                         | imaEP                                                                               |

### 3.5.5. Peti korak: Opis ad-hoc binarnih relacij

V petem koraku se podrobno definira vsako od ad-hoc relacij.

Naredimo tabelo ad-hoc binarnih relacij (Tabela 13), kjer poleg imena relacije in konceptov, ki jih relacija povezuje, določimo še nekaj metalastnosti relacij, kot so mestnost relacije (1:1, 1:N, M:N), matematične lastnosti relacije (refleksivnost, simetričnost, antisimetričnost, tranzitivnost ...) in morebitno obratno relacijo.

*Tabela 13: Tabela ad-hoc binarnih relacij.*

| Ime relacije               | Izvorni koncept                                                                                                                     | Ciljni koncept       | Kard. (max) | Mat. last. | Inverzna relacija |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------|----------------------|-------------|------------|-------------------|
| imaProceduro               | ŽivljenjskaSituacija                                                                                                                | Procedura            | 1           | -          | izvajaŽS          |
| imaStoritev                | ŽivljenjskaSituacija                                                                                                                | Storitev             | N           | -          | -                 |
| imaEStoritev               | Storitev                                                                                                                            | EStoritev            | 1           | -          | -                 |
| imaOdločbo                 | Storitev                                                                                                                            | Dokument             | 1           | -          | jeOdločba         |
| imaVlogo                   | Storitev                                                                                                                            | Dokument             | 1           | -          | -                 |
| imaPrilogo                 | Storitev                                                                                                                            | Dokument             | N           | -          | -                 |
| imaPonudnika               | Storitev                                                                                                                            | Ponudnik             | 1           | -          | -                 |
| imaPritožbo                | Storitev                                                                                                                            | Storitev             | 1           | -          | -                 |
| imaPonudnika               | EStoritev                                                                                                                           | Ponudnik             | 1           | -          | -                 |
| jeOdločba                  | Dokument                                                                                                                            | Storitev             | 1           | -          | imaOdločbo        |
| imaŽS                      | Uporabnik                                                                                                                           | ŽivljenjskaSituacija | N           | -          | -                 |
| imaDokument                | Uporabnik                                                                                                                           | Dokument             | N           | -          | -                 |
| imaStart                   | Procedura                                                                                                                           | EPStart              | 1           | -          | -                 |
| imaStop                    | Procedura                                                                                                                           | EPStop               | 1           | -          | jeStop            |
| izvajaŽS                   | Procedura                                                                                                                           | ŽivljenjskaSituacija | 1           | -          | imaProceduro      |
| imaEP                      | EPStart,<br>EPStoritev,<br>EPŽivljenjskaSituacija,<br>EPPodprocedura,<br>EPRazdruževalniIN,<br>EPZdruževalniIN,<br>EPZdruževalniALI | ElementProcedure     | N           | -          | -                 |
| imaEPYES                   | ElementProcedure                                                                                                                    | ElementProcedure     | 1           | -          | -                 |
| imaEPNO                    | ElementProcedure                                                                                                                    | ElementProcedure     | 1           | -          | -                 |
| jeStop                     | Stop                                                                                                                                | Procedura            | 1           | -          | imaStop           |
| sprožiStoritev             | EPStoritev                                                                                                                          | Storitev             | 1           | -          | -                 |
| sprožiŽivljenjskoSituacijo | EPŽivljenjskaStoritev                                                                                                               | ŽivljenjskaSituacija | 1           | -          | -                 |
| sprožiPodproceduro         | EPPodprocedura                                                                                                                      | Procedura            | 1           | -          | -                 |

Na podlagi prvih petih korakov konceptualizacije (4.5.1 – 4.5.5) lahko sedaj narišemo diagram UML z vključenimi informacijami o konceptih, njihovih atributih ter relacijah med koncepti in atributih teh relacij (Slika 27).

Potem ko smo naredili konceptualni model znanja s področja ŽS na najvišjem nivoju, ki smo ga tehnično gledano zapisali v obliki UML-diagrama razredov, je treba narediti še model reševanja vsake ŽS posebej. Pri tem je bistvena informacija, kako si posamezni elementi procedure sledijo oz. kakšen je vrstni red korakov pri reševanju ŽS. Na spodnji sliki (Slika 28) je predstavljen UML-diagram primerkov za ŽS »Poročim se«, ki je bil v obliki diagrama poteka predstavljen na eni od prejšnjih slik (Slika 15). Zaradi preglednosti so s sivo barvo obarvani liki, ki pripadajo razredu ElementProcedure. Opozoril bi še na poimenovanje primerkov razreda ElementProcedure na spodnji sliki. Imena so taka, kot jih je predlagal urejevalnik ontologij Protege, ko smo pozneje modelirali omenjeno proceduro. Zaradi primerljivosti slik (Slika 28 in Slika 33) so imena na obeh slikah identična.

```

classDiagram
    class Procedura {
        -ID
        -ČasZačetka
        -ČasKončanja
    }
    class ElementProcedure {
        -ID
        -ČasStart
        -ČasStop
    }
    class PonudnikStoritev {
        -ID
        -ime
        -kompetence
    }
    class EStoritev {
        -ID
        -ime
        -hyperpovezava
    }
    class ŽivljenjskaSituacija {
        -ID
        -ime
        -opis
        -ciljnaSkupina
        -organZaUrejanje
    }
    class Storitev {
        -ID
        -ime
        -pravnaPodlaga
        -takse
        -časIzvajanja
    }
    class Dokument {
        -ID
        -ime
        -datumIzdaje
    }
    class Uporabnik {
        -ID
        -imePriimek
        -datumRojstva
        -stan
        -državljanstvo
    }

    Procedura --> ElementProcedure : imaStart
    ElementProcedure --> Procedura : imaStop
    ElementProcedure --> ElementProcedure : imaEP
    ElementProcedure --> ElementProcedure : jeStop
    ElementProcedure --> PonudnikStoritev : sprožiStoritev
    PonudnikStoritev --> EStoritev : imaPonudnika
    EStoritev --> EStoritev : imaEStoritev
    ŽivljenjskaSituacija --> ElementProcedure : sprožiŽS
    ŽivljenjskaSituacija --> ŽivljenjskaSituacija : izvajaŽS
    ŽivljenjskaSituacija --> Storitev : imaStoritev
    Storitev --> PonudnikStoritev : imaPritožbo
    Storitev --> Dokument : jeDokument
    Dokument --> Dokument : imaDokument
    Uporabnik --> ŽivljenjskaSituacija : imaŽS
    Uporabnik --> Dokument : imaDokument
    
```

Diagrama procesa (EP) za prijavo poroke in pridobitev dovoljenja za poroko mladostnika. Diagram prikazuje zaporedje korakov, od prijave poroke do pridobitve dovoljenja, vključno s posredovanjem dokumentov in spreminjanjem osebnih podatkov.

```

graph TD
    In[ ] -- imaStart --> EPStart_3[EPStart_3]
    EPStart_3 -- imaEP --> EPRazdruzeva_IniIN_4[EPRazdruzeva IniiN_4]
    EPRazdruzeva_IniIN_4 -- imaEP --> EPRazdruzeva_IniALI_1[EPRazdruzeva IniiALI_1]
    EPRazdruzeva_IniIN_4 -- imaEP --> EPRazdruzeva_IniALI_14[EPRazdruzeva IniiALI_14]
    EPRazdruzeva_IniALI_1 -- imaEPNO --> EPZdruzevalni_ALI_3[EPZdruzevalni ALI_3]
    EPZdruzevalni_ALI_3 -- imaEP --> EPStoritev_2[EPStoritev_2]
    EPStoritev_2 -- imaStoritev --> Pridobite_dovoljenja[Pridobite dovoljenja za poroko mladostnika]
    EPRazdruzeva_IniALI_14 -- imaEPYES --> EPPodprocedura_15[EPPodprocedura_15]
    EPRazdruzeva_IniALI_14 -- imaEPNO --> EPPodprocedura_16[EPPodprocedura_16]
    EPPodprocedura_15 -- imaPodproceduro --> Pridobitev_dokumentov_1[Pridobitev dokumentov za tujca]
    EPPodprocedura_16 -- imaPodproceduro --> Pridobitev_dokumentov_2[Pridobitev dokumentov za tujca]
    EPPodprocedura_15 -- imaEP --> EPZdruzevalni_ALI_17[EPZdruzevalni ALI_17]
    EPPodprocedura_16 -- imaEP --> EPZdruzevalni_N_4[EPZdruzevalni N_4]
    EPZdruzevalni_ALI_17 -- imaEP --> EPZdruzevalni_N_4
    EPZdruzevalni_N_4 -- imaEP --> EPStoritev_5[EPStoritev_5]
    EPStoritev_5 -- imaStoritev --> Prijava_poroke[Prijava poroke]
    EPRazdruzeva_IniALI_6[EPRazdruzeva IniiALI_6] -- imaEP --> EPStoritev_5
    EPRazdruzeva_IniALI_6 -- imaEPNO --> EPZdruzevalni_ALI_8[EPZdruzevalni ALI_8]
    EPZdruzevalni_ALI_8 -- imaEP --> EPStoritev_7[EPStoritev_7]
    EPStoritev_7 -- imaStoritev --> Pritozba_na_prijavo_poroke[Pritozba na prijavo poroke]
    Sprememba_osebni_h_podatkov[Sprememba osebnih podatkov] -- imaZS --> EPZivljenjskaS_ituacija_11[EPZivljenjskaS ituacija_11]
    EPZivljenjskaS_ituacija_11 -- imaEP --> Razdruzevalni_ALI_10[Razdruzevalni ALI_10]
    EPZivljenjskaS_ituacija_11 -- imaEP --> EPZdruzevalni_ALI_12[EPZdruzevalni ALI_12]
    Razdruzevalni_ALI_10 -- imaEPNO --> EPZdruzevalni_ALI_12
    Razdruzevalni_ALI_10 -- imaEP --> EPStoritev_9[EPStoritev_9]
    EPZdruzevalni_ALI_12 -- imaEP --> EPStop_13[EPStop_13]
    EPStoritev_9 -- imaStoritev --> Pritozba_na_prijavo_poroke
    EPStop_13 -- jeStop --> Out[ ]
  
```

### 3.5.6. Šesti korak: Opis atributov primerkov

V šestem koraku se natančno določi vse attribute primerkov, ki se bili navedeni v Slovarju konceptov (Tabela 12). Pomembna odločitev pri tem koraku je, ali je nek atribut del znanja o področju ali bo le atribut v bazi podatkov neke aplikacije, ki uporablja znanje, zapisano v ontologiji. Tak primer je atribut datumIzdaje, ki je lahko atribut koncepta Dokument – v tem primeru bomo v ontologijo zapisovali vsako izdano odločbo. Lahko pa informacije o vsaki izdani odločbi prepustimo registru dokumentov, ki bi ga vodil tisti, ki ga to zanima, npr. Ministrstvo za javno upravo. Menim, da je slednja možnost lažja za obvladovanje, ker je bolj modularna, saj je v ontologiji zapisano le znanje, ki je relevantno za množico različnih vrst uporabnikov. Na ta način se seznam atributov močno skrči in je predstavljen v spodnji tabeli (Tabela 14), ki je rezultat tega koraka. Omenimo še, da nas v tej fazi zanimajo le najpomembnejši atributi in da lahko nabor atributov pozneje poljubno razširimo.

Tabela 14: Atributi primerkov.

| Ime atributa    | Koncept                                                                                    | Tip      | Merska enota | Natančnost | Obseg | Kardinalnost |
|-----------------|--------------------------------------------------------------------------------------------|----------|--------------|------------|-------|--------------|
| imePriimek      | Uporabnik                                                                                  | String   | -            | -          | -     | (0,N)        |
| datumRojstva    | Uporabnik                                                                                  | Date     | -            | day        | -     | (0,1)        |
| Stan            | Uporabnik                                                                                  | String   | -            | -          | -     | (0,1)        |
| Državljanstvo   | Uporabnik                                                                                  | String   | -            | -          | -     | (0,1)        |
| ime             | ŽivljenjskaSituacija<br>PonudnikStoritev<br>Dokument<br>Storitev<br>EStoritev<br>procedura | String   | -            | -          | -     | (1,1)        |
| opis            | ŽivljenjskaSituacija                                                                       | String   | -            | -          | -     | (0,1)        |
| ciljnaSkupina   | ŽivljenjskaSituacija                                                                       | String   | -            | -          | -     | (0,1)        |
| organZaUrejanje | ŽivljenjskaSituacija                                                                       | String   | -            | -          | -     | (0,1)        |
| pravnaPodlaga   | Storitev                                                                                   | String   | -            | -          | -     | (0,1)        |
| takse           | Storitev                                                                                   | Currency | EUR          | 0,01       | -     | (0,1)        |
| časIzvajanja    | Storitev                                                                                   | Time     | -            | -          | -     | (0,1)        |
| link            | EStoritev<br>PonudnikStoritev<br>Vloga                                                     | String   | -            | -          | -     | (0,1)        |
| Odgovornost     | ElementProcedure                                                                           | String   | -            | -          | -     | (0,1)        |
| Vprasanje       | EPRazdelilniALI                                                                            | String   | -            | -          | -     | (0,1)        |
|                 |                                                                                            |          |              |            |       |              |

### 3.5.7. Sedmi korak: Opis lastnosti razredov

V sedmem koraku se natančno določi vse attribute razredov, ki se bili navedeni v Slovarju konceptov (Tabela 12). Ta korak je podoben prejšnjemu, razlika je le v tem, da se v tem koraku določajo atributi razredov, ki pripadajo razredu, v katerem so določeni, ter se ne dedujejo na podrazrede in primerke. V našem primeru ne zaznavamo potrebe, da bi tovrstne lastnosti definirali, pa tudi OWL jih odsvetuje ter priporoča uporabo relacij in omejitev, ki jih sklepalniki lahko obdelajo, medtem ko atributi razredov (annotation properties) zanje niso razumljivi.

### 3.5.8. Osmi korak: Opis konstant

V osmem koraku se zapiše glavne konstante proučevanega področja. Rezultat tega koraka je tabela konstant, kjer se navedejo imena konstant, njihova vrednost, tip vrednosti in merska enota. V tem trenutku ne vidimo potrebe po nobeni konstanti, zato tudi nobene nismo definirali.

### 3.5.9. Deveti korak: Opis formalnih aksiomov

V devetem koraku se zapišejo formalni aksiomi, ki so po definiciji vedno resnične logične izjave in ki se uporabljajo za zapis omejitev oz. robnih pogojev. Če bi jim kakšen podatek v zbirki znanja nasprotoval, bi bila zbirka znanja nekonsistentna, s čimer predstavljajo pomemben koncept pri gradnji večjih težkokategornih ontologij z mnogo povezav in znanja.

V spodnji tabeli so predstavljeni trije aksiomi, ki lahko služijo za nadzor konsistentnosti znanja v ontologiji. Pozneje se lahko kadarkoli dodajo novi aksiomi.

Tabela 15: Formalni aksiomi v ontologiji življenjskih situacij.

| Ime aksioma                | ŽS državljanov                                                                                               | Start Procedure                                                                                                                                 | Ponudnik e-storitve                                                                                                                                                                        |
|----------------------------|--------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Opis v naravnem jeziku     | Državlani lahko rešujejo le življenjske situacije državljanov.                                               | Element Start je popolnoma na začetku vsake procedure<br>Procedura (noben drug Element ne kaže nanj).                                           | Ponudnik e-storitve je isti kot ponudnik običajne storitve.                                                                                                                                |
| Zapis z logiko prvega reda | <code>forall(?X, ?Y) ( [Državljan Uporabnik] (?X) and [imaŽS] (?X, ?Y) =&gt; [ŽSDržavljana ŽS] (?Y) )</code> | <code>forall(?X, ?Y) ( [ElementProcedure] (?X) and [ElementProcedure] (?Y) and [imaEP] (?X, ?Y) =&gt; ![EPStart ElementProcedure] (?Y) )</code> | <code>forall(?X, ?Y, ?Z) ( [Storitev] (?X) and [EStoritev] (?Y) and [PonudnikStoritev] (?Z) and [imaPonudnika] (?X, ?Z) and [imaEStoritev] (?X, ?Y) =&gt; [imaPonudnika] (?Y, ?Z) )</code> |
| Koncepti                   | Državljan<br>ŽSDržavljana                                                                                    | ElementProcedure                                                                                                                                | Storitev<br>EStoritev<br>PonudnikStoritev                                                                                                                                                  |
| Atributi                   | -                                                                                                            | -                                                                                                                                               | -                                                                                                                                                                                          |
| Relacije                   | imaŽS                                                                                                        | imaEP                                                                                                                                           | imaPonudnika<br>imaEStoritev<br>imaPonudnika                                                                                                                                               |
| Spremenljivke              | ?X, ?Y                                                                                                       | ?X, ?Y                                                                                                                                          | ?X, ?Y, ?Z                                                                                                                                                                                 |

### 3.5.10. Deseti korak: Opis pravil

V desetem koraku se določijo pravila, ki jih bomo, podobno kot aksiome, potrebovali pri izvajanju preverjanja in sklepanja. Podobno se naredi tabela pravil, pri čemer se pravila običajno podajajo v obliki *if <pogoji> then <posledice>*. Pri aksiomih podamo nekaj, kar vedno velja, pri pravilih pa podamo pogoje, pri katerih je nekaj določeno, npr. nek atribut zavzame neko vrednost pri nekih pogojih. V spodnji tabeli (Tabela 16) je predstavljen primer pravila s področja ŽS:

Tabela 16: Pravilo v ontologiji življenjskih situacij.

|                                   |                                                                                                                                                                                                                                                                                             |
|-----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Ime pravila</b>                | Zaključek storitve                                                                                                                                                                                                                                                                          |
| <b>Opis v naravnem jeziku</b>     | Storitev, katere sprožitev je zahteval ElementStoritev znotraj izvajanja Procedure, je končana, ko je generiran dokument, ki je odločba dotične storitve. Informacija se zapiše kot vrednost atributa ČasStop, ki dobi enako vrednost, kot jo ima atribut datumIzdaje določenega dokumenta. |
| <b>Zapis z logiko prvega reda</b> | <pre> if (   [ElementStoritev] (?X) and   [Storitev] (?Y) and   [Dokument] (?Z) and   [imaStoritev] (?X, ?Y) and   [imaOdločbo] (?Y, ?Z) ) then (   [časStop] (?X, [datumIzdaje] (?Z)) ) </pre>                                                                                             |
| <b>Koncepti</b>                   | ElementStoritev<br>Storitev<br>Dokument                                                                                                                                                                                                                                                     |
| <b>Atributi</b>                   | časStop<br>datumIzdaje                                                                                                                                                                                                                                                                      |
| <b>Relacije</b>                   | imaStoritev<br>imaOdločbo                                                                                                                                                                                                                                                                   |
| <b>Spremenljivke</b>              | ?X, ?Y, ?Z                                                                                                                                                                                                                                                                                  |

### 3.5.11. Enajsti korak: Opis primerkov

Enajsti korak je izbirni in predstavlja opisovanje ali vnašanje podatkov o primerkih. Primerki običajno niso del ontologije, pač pa konkretne zbirke znanja, vendar se včasih izkaže, da je primerno, da se kakšen primerek definira tudi v ontologiji sami.

Pri razvoju ontologije ŽS smo se odločili, da bomo proceduro reševanja konkretne ŽS modelirali na nivoju primerkov, zato bi bilo smiselno, da navedemo primerke koncepta ElementProcedure, s katerimi zapišemo neko ŽS. Ker pa je ta korak podrobneje predstavljen v nadaljevanju (poglavje 3.6.3), ga na tem mestu ne bomo opisovali.

## 3.6. FORMALIZACIJA MODELA

Namen te aktivnosti je pretvorba konceptualnega modela v formalni oz. polovično izvedljiv model, ki se ga v naslednji aktivnosti zapiše s primernim jezikom. Z uporabo primerne orodja se ta aktivnost zreducira na prenos konceptualnega modela v orodje (npr. Protege), če se to ni storilo že v fazi konceptualizacije.

Na podlagi spoznanj v prvem poglavju smo uporabili urejevalnik ontologij Protege, in sicer verzijo, ki podpira deskriptivne logike in omogoča zapis znanja z jezikom OWL.

Preden začnemo s prenašanjem konceptualnega modela v Protege, se moramo odločiti o poimenovanju, ki ga bomo uporabljali v ontologiji. Imena razredov, relacij ali atributov sicer ne vplivajo na samo semantiko znanja, zapisanega v ontologiji, so pa v pomoč tistemu, ki ontologijo ureja ali uporablja, ker že intuitivno opisujejo sam koncept. Naš dogovor je naslednji:

- Razrede bomo pisali z veliko začetnico, povezave in attribute (ki so v bistvu povezave na podatke) pa z malo začetnico (npr. Storitev in imaOdločbo).



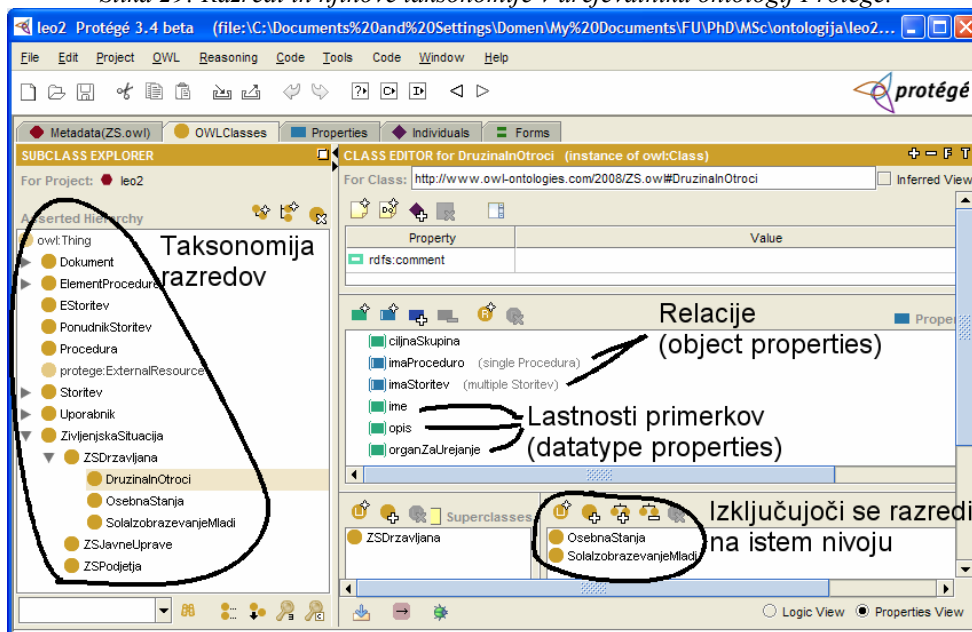
- Če ime vsebuje več besed, jih bomo pisali skupaj, to je brez presledka, pri čemer bo vsaka naslednja beseda pisana z veliko začetnico (npr. ŽivljenjskaSituacija).
- Uporabljali bomo slovenske besede brez posebnih slovenskih črk (ČŠŽ).
- Kjer je potrebno, bomo za podrazred uporabljali na začetku imena akronim nadrazreda (npr. ZSDrzavljana je podrazred razreda ŽivljenjskaSituacija).

### 3.6.1. Razredi in njihove lastnosti

Najprej vnesemo razrede in njihove taksonomije oz. podrazrede teh razredov (Slika 29). Privzeta nastavitve v Protegeju je, da ne loči razredov med seboj, kar pomeni, da je vsak primerik kateregakoli razreda lahko obenem primerik vsakega drugega razreda. Če torej želimo, da se dva razreda izključujeta, je treba to eksplicitno označiti (disjoint classes).

OWL sicer omogoča taksonomijo razredov (classes) ter opisovanje njihovih lastnosti (prek annotation properties), ki sta uporabnikom baz podatkov ter modeliranja UML intuitivno že poznana, vendar se pravo ontološko modeliranje OWL zakonitosti področja izvaja prek omejevanja (property restrictions), ki je namenjeno omejevanju lastnosti primerikov, ki pripadajo določenemu razredu. Omejitve se v OWL delijo na Quantifier Restrictions, Cardinality Restrictions in hasValue Restrictions (Horridge et al., 2004). Nad tako zapisanim znanjem potem izvajamo logično sklepanje in preverjanje s sklepalniki, kot so RACER in podobni. Ker pa takšno modeliranje v tem trenutku presega namen tega dela, smo pri opisovanju lastnosti uporabili predvsem lastnosti primerikov (datatype properties) (Slika 29).

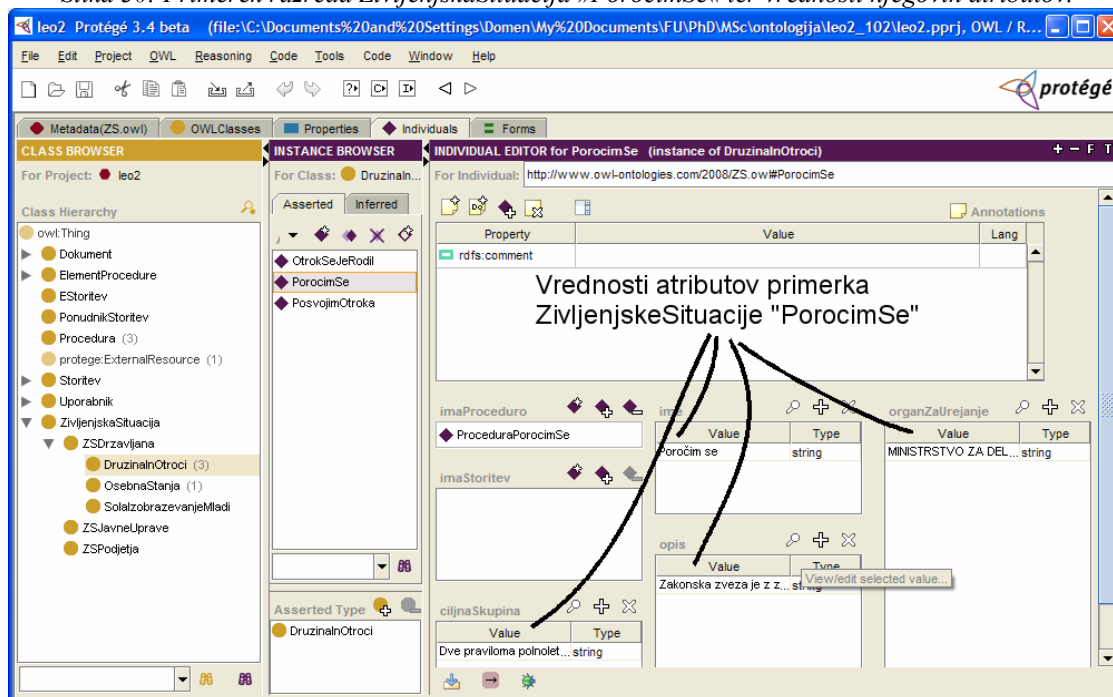
Slika 29: Razredi in njihove taksonomije v urejevalniku ontologij Protege.



Vrednosti atributov primerikov se vnesejo šele, ko definiramo primerik. Na sliki (Slika 30) je predstavljen vnos novega primerka razreda ŽivljenjskaSituacija, kjer se vidi, da urejevalnik Protege takoj ponudi vnos informacij, ki jih določa ontologija, kot je opis ŽS. Vrednosti atributov primerka ŽivljenjskaSituacija »PorocimSe« so bile pridobljene na

portalu e-uprava<sup>57</sup> (Slika 18). Na ta način smo zapisali nekaj najpomembnejših informacij, ki se jih za posamezno ŽS dobi na portalu e-uprava.

Slika 30: Primerek razreda ŽivljenjskaSituacija »PorocimSe« ter vrednosti njegovih atributov.



### 3.6.2. Relacije

Nato smo vnesli vse relacije, ki smo jih identificirali v koraku 3.5.3. V OWL-ju se imenujejo objektne lastnosti (angl. object properties) in so relacije med dvema primerkom v nasprotju s podatkovnimi lastnostmi (angl. datatype properties), ki so relacije med posameznikom in neko vrednostjo (angl. XML Schema Datatype value ali RDF literal-om).

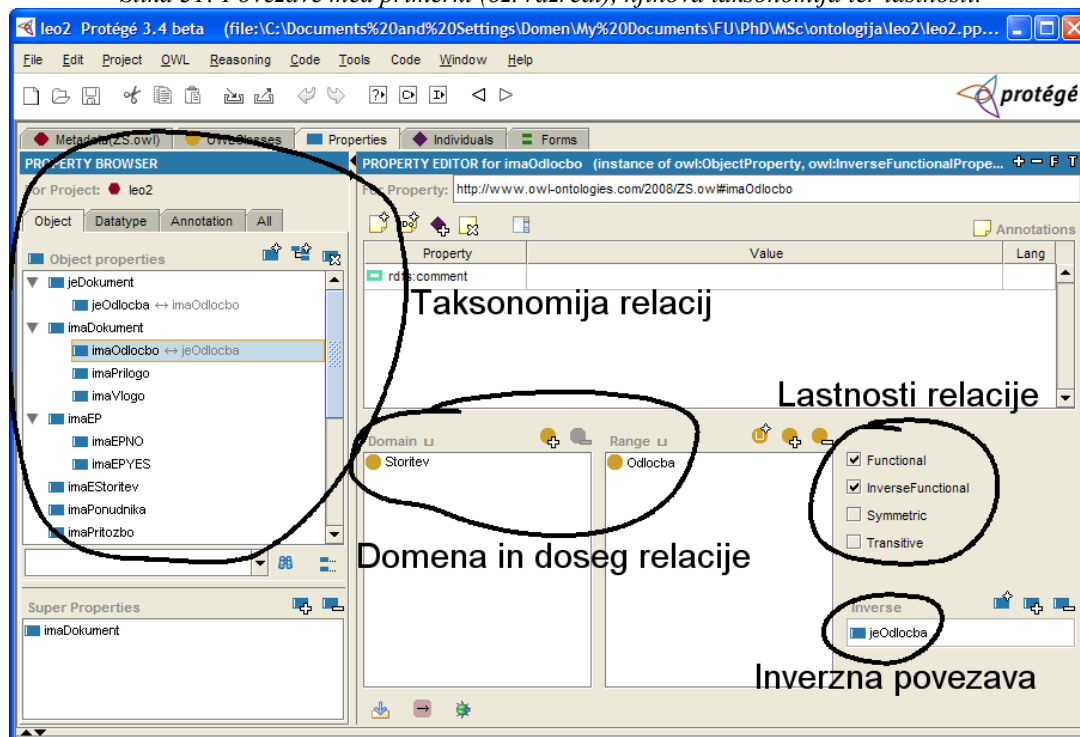
OWL dopušča oz. omogoča hierarhijo relacij, kot to velja za razrede. Na spodnji sliki (Slika 31) se vidi, da ima relacija imaDokument tri podrelacije: imaOdločbo, imaPrilogo in imaVlogo.

Za vsako relacijo se lahko določijo preddefinirane lastnosti (Slika 31): Functional, InverseFunctional, Symetric, Transitive in inverzna relacija. Functional pomeni, da je kardinalnost povezave 1, kar v primeru imaOdločbo pomeni, da ima Storitev samo eno odločbo in če imamo primerek Storitve, ki ima dve relaciji, ki povezujeta Storitev z dvema dokumentoma, potem bo sklepalni mehanizem zaključil, da gre za en in isti dokument. InverseFunctional pomeni, da je kardinalnost inverzne relacije 1. Symetric pomeni, da je relacija simetrična, in Transitive, da je relacija tranzitivna, vendar takih primerov v naši ontologiji zaenkrat ni. Inverzna relacija neke relacije je tista, ki opisuje obraten odnos, npr. imaOdločbo opisuje to, da ima Storitev Odločbo, v obratnem primeru pa relacija jeOdločba opisuje to, da je Dokument odločba neke Storitve.

<sup>57</sup> <http://e-uprava.gov.si/e-uprava/dogodkiPrebivalci.euprava?zdid=1179&sid=191>

Pri domeni in obsegu je treba povedati, da jih v OWL-ju ne smemo obravnavati kot omejitve (constraints), ki se jih preverja oz. one omejujejo definiranje povezav. Obravnavati jih moramo kot trditve (axioms), ki pridejo v poštev šele v fazi sklepanja. Tehnično to pomeni, da nam OWL oz. Protege dovoli definirati npr. povezavo imaOdlocbo med primerkom razreda Storitev in primerkom razreda EStoritev. Nesmiselnost oz. neusklajenost znanja bo ugotovil šele sklepalni mehanizem, pa še to samo pod pogojem, da smo eksplicitno definirali, da sta razreda EStoritev in Dokument disjunktna, kar smo v našem primeru tudi storili.

Slika 31: Povezave med primerki (oz. razredi), njihova taksonomija ter lastnosti.



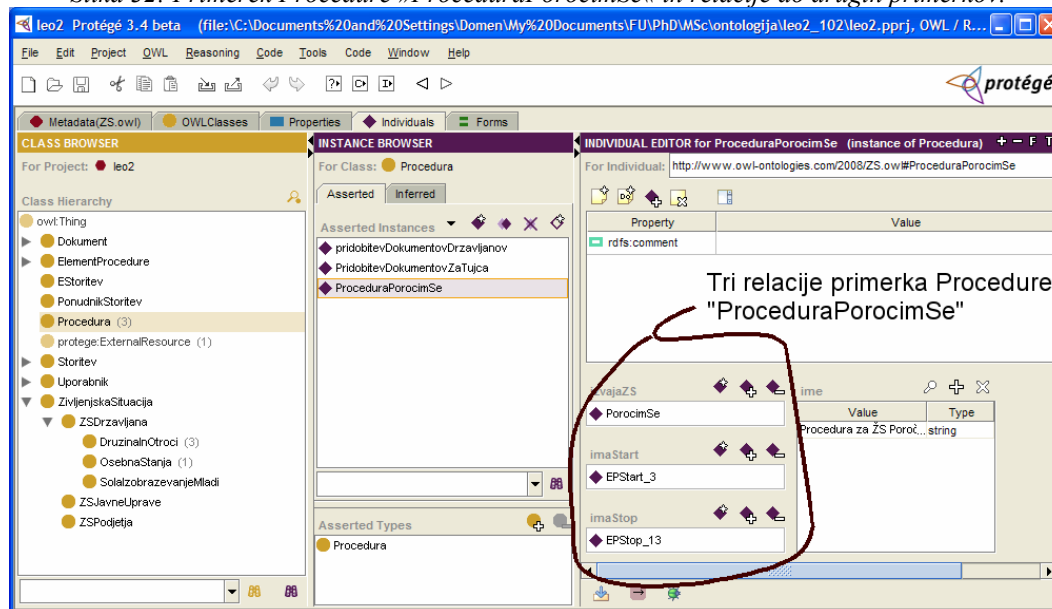
### 3.6.3. Procedura za reševanje življenjske situacije »PorocimSe«

Ko smo vnesli širše znanje o področju ŽS, se lotimo še zapisovanja znanja o konkretnih ŽS, ki ga bomo pokazali na primeru ŽS »PorocimSe«. Naš namen je, da bi v ontologijo prenesli diagram poteka za dotično ŽS z vsemi relevantnimi informacijami. Ker smo si v koraku 3.5.5 pripravil UML-model diagrama poteka za to ŽS, ga bomo sedaj enostavno prenesli v ontologijo. Znanje o poteku reševanja ŽS se ne spreminja z uporabo tega znanja, zato je logično, da je to del ontologije znanja. Proceduro reševanja posamezne ŽS bi lahko modelirali na nivoju razredov, vendar smo se odločili, da jo bomo na nivoju primerkov, ker je to v Protegeju veliko lažje. Lažje je namreč vnašati veliko primerkov enega razreda in ene vrste povezave, kot pa ustvarjati veliko razredov z veliko vrstami povezav. Kaj modelirati na nivoju primerkov in kaj na nivoju razredov, ni trivialno vprašanje in se razlikuje od primera do primera (Noy in McGuinness, 2001).

Celoten potek reševanja ŽS »PorocimSe« z vsemi povezavami na relevantne Storitve, Podprocedure ter povezane ŽivljenjskeSituacije smo vnesli v ontologijo v približno dveh urah. Pri tem nam je bila v veliko pomoč funkcija jezika OWL »TODO list«, ki je v bistvu preddefinirana lastnost vsakega objekta v ontologiji. Ko za kakšen objekt (primerek česarkoli, razred, relacijo ...) v nekem trenutku ontologist (v tem primeru je bil to avtor

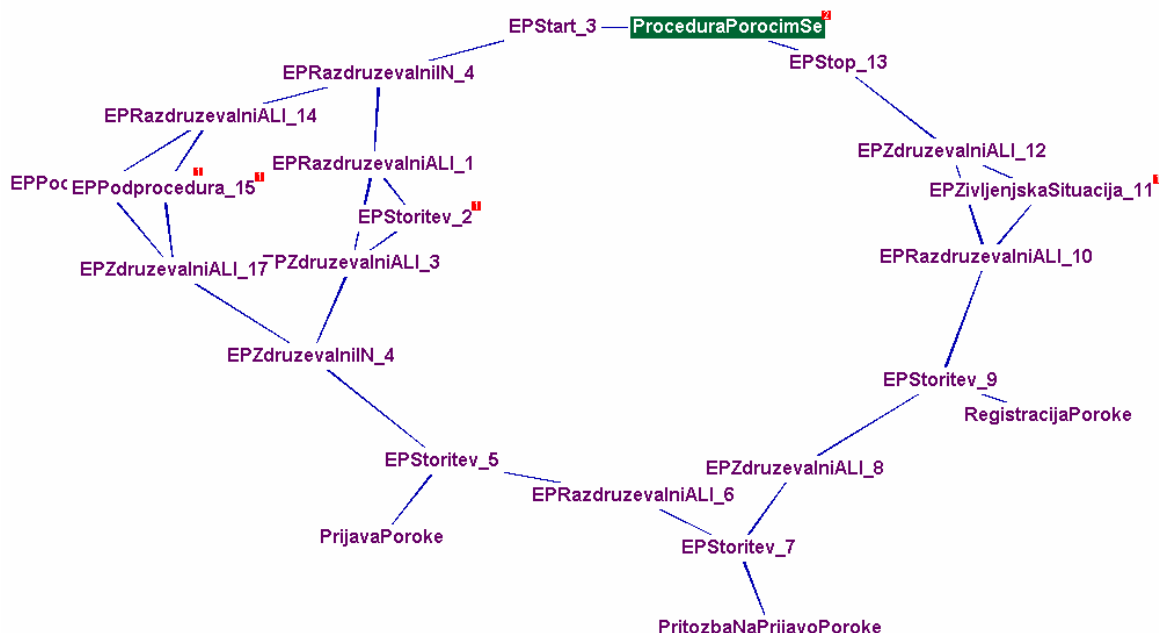
pričujočega magistrskega dela) nima časa določiti vsega potrebnega, enostavno vključi to lastnost. Pozneje lahko prikliče spisek vseh objektov, ki imajo vključeno to lastnost, in dokonča svoje delo. Na spodnji sliki (Slika 32) so prikazane relacije med primerkom Procedure »ProceduraPorocimSe«, ki služi zapisu znanja o proceduri reševanja ŽS »PorocimSe«.

Slika 32: Primerek Procedure »ProceduraPorocimSe« in relacije do drugih primerkov.



Procedura reševanja posamezne ŽS vključuje precej elementov oz. primerkov koncepta ElementProcedure. Da bi si pregledno prikazali primerke in povezave med njimi, ki so pomembne za proceduro reševanja posamezne ŽS, ima Protege nekaj vtičnikov (plug-in). TGViz omogoča poljubno dodajanje, skrivanje in definiranje novih elementov na grafu. Za nas pa je uporabna predvsem možnost hitrega vizualnega ogleda povezav, še posebej tistih, ki opisujejo procedure, ker so lahko tudi zelo razvejane. Na spodnji sliki (Slika 33) je prikazana procedura za reševanje ŽS »PorocimSe«, ki je bila na drugačen način prikazana že v prejšnjih poglavjih (Slika 15 in Slika 28). Rdeči kvadratici nad imenom elementa opozarjajo, da ima element še nekaj relacij, ki pa trenutno niso prikazane.

Slika 33: Vizualizacija procedure »ProceduraPorocimSe« s pomočjo vtičnika TGViz za urejevalnik ontologij Protege-OWL.



### 3.7. IMPLEMENTACIJA MODELA

V aktivnosti implementacije zakodiramo formalni model, razvit v prejšnji aktivnosti, v računalniku razumljiv model s pomočjo kakšnega od jezikov za predstavitev znanja, ki so bili predstavljeni v enem od prejšnjih poglavij (Poglavje 2.2).

Mi smo za zapis znanja izbrali jezik OWL. Urejevalnik ontologij Protege uporablja za zapis formalnega modela svoj lasten jezik. Z vtičnikom za OWL dobimo razširjeno različico Protege-OWL<sup>58</sup>, ki omogoča avtomatično prevajanje iz Protegejevega zapisa v jezik OWL (Knublauch et al., 2004). Zaradi uporabnosti omenjenih orodij je aktivnost implementacije dokaj trivialno opravilo.

Celotna ontologija ŽS, zapisana v jeziku OWL, je objavljena na spletu, kjer je prosto dostopna prek povezav:

<http://lifeeventontology.googlepages.com/leo.owl> ,

[http://protegewiki.stanford.edu/index.php/Protege\\_Ontology\\_Library#OWL\\_ontologies](http://protegewiki.stanford.edu/index.php/Protege_Ontology_Library#OWL_ontologies) .

Znanje, zapisano v ontologiji, je namenjeno v grobem dvema vrstama uporabnikov: ljudem in računalniškim aplikacijam. Ljudje lahko dostopamo do znanja neposredno prek urejevalnikov ontologij, kot je Protege, kjer se s klikanjem po objektih ontologije prebijemo do znanja, ki ga iščemo, ali pa uporabimo poizvedovalni jezik SPARQL (SPARQL Protocol and RDF Query Language), ki je od začetka 2008 tudi uradno priporočilo konzorcija W3C. Lahko pa uporabimo tudi semantične brskalnike, kot je Swoogle<sup>59</sup>. Računalniške aplikacije dostopajo do znanja prek vmesnikov, ki bi jih uporabili tudi, če bi želeli predstaviti znanje na spletu v človeku razumljivi obliki. Pri tem

<sup>58</sup> <http://protege.stanford.edu/>

<sup>59</sup> <http://swoogle.umbc.edu/>

je glavna ideja v transformaciji datoteke .owl v zapis .html s pomočjo šablon XSLT (EXtensible Stylesheet Language) (Laclavik et al., 2005).

### **3.8. VREDNOTENJE ONTOLOGIJE**

V aktivnosti vrednotenje izvajamo tehnološko presojo ontologije, programske opreme in dokumentacije. Ontologija bi morala biti namreč testirana, preden se začne uporabljati. Evalvacija se deli na dva dela: tehnična, ki jo izvedejo razvijalci, ter uporabniška, ki jo izvedejo uporabniki (Gomez-Perez et al., 1995) (Garcia, 2008). Ker je naša ontologija še v fazi predloga, vrednotenje ni bilo popolno in ima bolj namen pokazati nekaj metod vrednotenja.

Najprej smo se lotili tehničnega vrednotenja. Urejevalnik Protege-OWL že sam po sebi ponuja nekaj testiranja, ki smo ga uporabili med samo gradnjo ontologije. Nato smo uporabil WonderWeb OWL Validator<sup>60</sup>, ki nam je sporočil naslednje:

Untyped Individual: [http://www.w3.org/2002/07/owl#Procedura\\_2](http://www.w3.org/2002/07/owl#Procedura_2)

V ontologiji je bil očitno zapis nekega razreda, ki ni imel starša, kar je verjetno posledica kakšnega hrošča v urejevalniku.

Vrednotenje s strani uporabnikov pride na vrsto pozneje in v tem trenutku presega namen tega dela.

### **3.9. VZDRŽEVANJE ONTOLOGIJE**

V aktivnosti vzdrževanje popravljamo in posodabljammo ontologijo, če je to potrebno. Ta korak pride na vrsto pozneje, ko se bo ontologija začela uporabljati.

### **3.10. UPORABA ONTOLOGIJE**

Ker je ontologija trenutno v fazi prototipa, lahko o njeni uporabi govorimo le na hipotetičnem nivoju oz. si jo le predstavljamo. Sama aktivnost uporabe je cilj nadaljnjega raziskovanja. V nadaljevanju je prikazano pokrivanje informacijskih potreb oz. možnost uporabe znanja, zapisanega v ontologiji, s strani različnih tipov uporabnikov (Tabela 17).

---

<sup>60</sup> <http://phoebe.cs.man.ac.uk:9999/OWL/Validator>

Tabela 17: Pokrivanje informacijskih potreb različnih uporabnikov s strani ontologije življenjskih situacij:

| Tip uporabnika                   | Informacijske potrebe                                                                                                                                  | Znanje, zapisano v ontologiji ŽS                                                                                                                                                                                                                                                       |
|----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Državljeni                       | Seznam storitev za njihovo ŽS, opis storitev, vrstni red njihovega sprožanja ter kontaktni naslovi pristojnih organov.                                 | Vse informacije so dostopne kot atributi konceptov (datatype properties). Do poteka reševanja, ki upošteva vse relevantne okoliščine uporabnika, pride uporabnik prek povezav (object properties) med primerki koncepta ElementProcedure.                                              |
| Uradniki javne uprave            | Enako kot pri državljanih, le da pogosto potrebujejo podrobnejše informacije o točno določenih objektih v ontologiji.                                  | Velja enako kot za državljana, le da bi uporabniki lahko uporabljali bolj prilagojene vmesnike ali celo sam urejevalnik ontologij ter že razvite vmesnike za pridobivanje informacij iz obstoječih evidenc in registrov.                                                               |
| Razvijalci elektronskih storitev | Poleg ostalih informacij so za njih zanimive informacije o obstoječih dogovorih/standardih glede komunikacije med e-servisi in procedure reševanja ŽS. | Procedure reševanja ŽS so vključene v ontologijo. Kar se tiče interoperabilnostnih standardov in dogovorov pa ontologija ponuja centralen pogled na že razvite servise in primere delujoče prakse, ki lahko služijo kot napotilo za usmerjeno iskanje interoperabilnostnih informacij. |
| Aplikacije elektronskih storitev | Zanje je zanimiva predvsem procedura reševanja ŽS in povezave na relevantne dokumente (vloge in odločbe).                                              | Ker je vse znanje zapisano z jezikom OWL, ki je postal uradno priporočilo za gradnjo semantičnega spleta, lahko razvijalci zgradijo standardne odjemalce znanja na standardnih tehnologijah, ki lahko kadarkoli dostopajo do znanja, zapisanega v ontologiji.                          |
| Zakonodajalci                    | Potrebujejo znanje kot vsi predhodni uporabniki, ki jim ga tudi dostavljajo in interpretirajo.                                                         | Če imajo vsi predhodni uporabniki omogočen dostop do znanja v ontologiji, ga lahko na enostaven način posredujejo zakonodajalcem.                                                                                                                                                      |

Poleg informacij o posameznih konceptih oz. objektih (dokument, storitev ...) omogoča razvita ontologija aktivno vodenje uporabnika pri reševanju njegove ŽS z upoštevanjem specifičnih okoliščin, v katerih je le-ta. Pokažimo to na konkretnem primeru. Oseba A se želi poročiti. S pomočjo taksonomije ŽS ali iskalnika ŽS, ki vrne ŽS na podlagi njenega opisa, ki je tudi shranjen v ontologiji, bi uporabnik izbral ŽS »Poročim se«, ki ima povezavo na Proceduro za njeno reševanje. Procedura ima neposredno povezavo s prvim elementom procedure, ta pa naprej z drugim. Tretji element procedure je razdruževalni element, ki ima dve povezavi in obe se uporabita, ker gre za razdrževalni IN. Prva povezava nato privede do vprašanja »Ali je kandidat za poroko mladoletna oseba?«. Če je oseba mladoletna, ga ontologija vodi do storitve PridobitevDovoljenjaZaPorokoMladoletneOsebe, če pa ni mladoleten, ontologija ponudi povezavo, ki omenjeno storitev preskoči in se z njo ali z njo povezanimi dokumenti ne ukvarja več. Ontologija na tak način vodi do zadnjega elementa procedure, ki se imenuje EPStop in nedvoumno pomeni zaključek reševanja določene ŽS.

## SKLEP

### POVZETEK REZULTATOV DELA

V magistrskem delu smo podali predlog ontologije življenjskih situacij (ŽS) in postopek gradnje le-te.

Razvili smo prototip ontologije ŽS, ki se od običajne baze podatkov razlikuje v tem, da:

- je znanje o področju ločeno od operativnega znanja primerov uporabe;
- je znanje o področju dostopno navzven in ne le znotraj baze podatkov;
- zapis znanja upošteva standarde semantičnega spleta in je tako dostopno prek spleta;
- je znanje o področju razumljivo semantičnim računalniškim agentom.

V ontologiji smo zajeli znanje o področju ŽS v okviru slovenske e-uprave, ki vključuje:

- taksonomijo in opis ŽS;
- procedure reševanja ŽS;
- enostavne taksonomije in opise storitev, e-storitev, ponudnikov storitev, uporabnikov in dokumentov.

Pri razvoju smo namenoma uporabili najmanjši možen nabor različnih, vendar standardnih tehnologij ter najmanjši možen nabor konceptov in povezav, ker smo zasledovali cilj, da naj bo vse čimbolj enostavno. Nabor konceptov in povezav se lahko kadarkoli razširi, če se izkaže, da je to potrebno.

V primerjavi z ostalimi pristopi k modeliranju področja ŽS ima naš pristop več prednosti:

- znanje o področju ŽS in proceduri reševanja ŽS je zapisano z enim samim jezikom (v jezik OWL se prenese znanje, zbrano v tabelah in diagramih poteka);
- zapis znanja podpira aktivno vodenje uporabnika pri reševanju njegove ŽS, pri čemer upošteva specifične okoliščine, v katerih je trenutno uporabnik;
- vse kombinacije okoliščin so modelirane na enem mestu in samo enkrat (ta lastnost se v mnogih ostalih pristopih izključuje s prejšnjo).

### OVREDNOTENJE REZULTATOV DELA GLEDE NA HIPOTEZE

Na začetku dela smo postavili tri hipoteze, ki jih bomo na tem mestu potrdili ali ovrgli.

Prva hipoteza: Nabor konceptov, ki za različne uporabnike znanja o ŽS zadovoljivo opišejo ŽS ter postopke slovenske javne uprave, je omejen. Tabela 17 prikazuje primer uporabe ontologije ŽS in njeno sposobnost pokrivanja informacijskih potreb za različne tipe uporabnikov. Na podlagi v tabeli predstavljenih primerov lahko zaključimo, da je prva hipoteza potrjena.

Druga hipoteza: Izdelan konceptualni model se da zapisati kot ontologija in implementirati v obliki, ki jo lahko obravnavajo obstoječa računalniška orodja, temelječa na spletnih in semantičnih tehnologijah. Konceptualni model smo zapisali z jezikom OWL, ki temelji na semantičnih in spletnih standardih, poleg tega pa je priporočilo konzorcija W3C. OWL



omogoča tudi zapis oz. predstavitev znanja v obliki težkokategorne ontologije, saj smo uporabili različico jezika, ki temelji na deskriptivnih logikah OWL-DL. Zato lahko zaključim, da je druga hipoteza potrjena.

Tretja hipoteza: Razvita ontologija omogoča aktivno vodenje uporabnika pri reševanju njegove ŽS z upoštevanjem specifičnih okoliščin, v katerih je. V poglavju 3.10 je predstavljen primer aktivnega vodenja uporabnika. Vse informacije za prilagajanje reševanja ŽS uporabnikovim specifičnim okoliščinam so zapisane v sami ontologiji. Zato lahko zaključimo, da je tretja hipoteza potrjena.

## **OVREDNOTENJE PRISPEVKA K ZNANOSTI**

Pričujoče magistrsko delo spada na znanstveno področje predstavitve znanja, modeliranja poslovnih procesov in na področje uporabe informacijskih orodij v slovenski javni upravi. Prvi prispevek k znanosti je analiza in primerjava pristopov k modeliranju ŽS. Drugi prispevek k znanosti je analiza možnih pristopov k predstavitvi znanja ter izbira najprimernejšega od njih za modeliranje področja ŽS. Tretji prispevek k znanosti pa je sam prototip ontologije ŽS, ki podpira aktivno vodenje uporabnika pri reševanju njegove ŽS ob upoštevanju specifičnih okoliščin.

## **SMERNICE ZA NADALJNJE DELO**

V tem delu je predstavljen razvoj prototipa ontologije ŽS, ki deluje na omejenem obsegu znanja. Zato predlagamo nadaljevanje dela v treh korakih: (1) pristop obširneje ovrednotiti in dodatno opravičiti razvoj polno zmogljive ontologije; (2) ontologijo dejansko napolniti s potrebnim znanjem in razviti potrebno orodje za lažje delo z njo; (3) ontologijo predati v dejansko uporabo.

V prvem koraku predlagamo naslednje aktivnosti:

- primerjava tega pristopa z ostalimi obstoječimi pristopi k modeliranju procesov in storitev (BPMN, OWL-S, SWS ...);
- testiranje, koliko časa bi vzelo modeliranje vseh ŽS ter izračun stroškov za razvoj polno zmogljive ontologije;
- testiranje prihranka za vse tipe uporabnikov;
- izračun upravičenosti razvoja polno zmogljive ontologije.

V drugem koraku predlagamo naslednje aktivnosti:

- zagotoviti možnost grafične prezentacije diagrama poteka (plug-in za Protege);
- avtomatično generiranje OWL kode za diagram poteka iz grafičnega modela (grafični model izvoziti iz VISIO-a v XML);
- dodati znanje, ki je potrebno za omogočanje semantične povezljivosti e-storitev;
- izdvojiti ontologijo diagrama poteka;
- zgraditi ontologijo javne uprave (ponudniki storitev, dokumenti, storitve);
- povezati vse tri ontologije med seboj ter z ostalimi (vrhnjimi in domenskimi) ontologijami;
- napolniti ontologijo z znanjem;
- rešiti problem večjezičnosti (verjetno prek annotation properties);

- izgradnja potrebnih vmesnikov za omogočanje dejanske uporabe različnim tipom uporabnikov.

V tretjem koraku predlagamo naslednje aktivnosti:

- ontologijo predati v uporabo vsem tipom uporabnikov;
- analizirati težave in prihranke vseh tipov uporabnikov ter nadalje ukrepati;
- vključiti ontologijo v nacionalni interoperabilnostni okvir.

## LITERATURA IN VIRI:

- [1] Adam, O., Werth, D., Zangl, F. (2003). Conceiving and implementing pan-European integrated public services. V: *Proceedings of EGOV 2003* (DEXA 2003).
- [2] Ahmad, M.N. in Colomb, R.M. (2007). Managing ontologies: a comparative study of ontology servers. V ADC '07: Proceedings of the eighteenth conference on Australasian database, str. 13–22, Australian Computer Society, Inc.
- [3] Alavi, M. in Leidner, D.E. (2001). Review: Knowledge management and knowledge management systems: Conceptual foundations and research issues. *MIS Quarterly*, 25(1), str. 107–136.
- [4] Apostolou, D., Stojanovic, L., Lobo, T., P., Thoenssen, B. (2005). Towards a Semantically-Driven Software Engineering Environment for eGovernment. M. Bohlen et al. (Eds.): *TCGOV 2005*, LNAI 3416, pp. 157–168.
- [5] Arpirez, A., Corcho, J., C., Fernandez-Lopez, O., Gomez-Perez, M. (2003). WebODE in a nutshell. *AI Magazine*.
- [6] Atkinson, R.D. in Castro, D. (2008). Digital Quality of Life, The Information Technology and Innovation Foundation, str. 137–145. Najdeno 15. 12. 2008 na spletu: <http://www.itif.org/files/DQOL-14.pdf>.
- [7] Baader, F. in Hollunder, B. (1991). KRIS: Knowledge Representation and Inference System. *ACM SIGART Bulletin* 2(3). *Special issue on implemented knowledge representation and reasoning systems*. pp. 8–14.
- [8] Baader, F., Lutz, C., Suntisrivaraporn, B. (2006). CEL – A Polynomial-time Reasoner for Life Science Ontologies. *3rd International Joint Conference on Automated Reasoning (IJCAR'06)*. Springer-Verlag.
- [9] Bekkers, V. (2005). The Governance of Back Office Integration in E-Government: Some Dutch Experiences. *EGOV 2005*: 12–25.
- [10] Bent, S., Kernaghan, K. in Marson, B. D. (1999). *Innovations and Good Practices in Single-Window Service*, Canadian Centre for Management Development.
- [11] Berčič, B. (2006). *Modeliranje delno strukturiranih odločitvenih situacij na podlagi modela normativnih aktov*. Doktorska disertacija, Ljubljana.
- [12] Berčič, B., Vintar, M. (2002). Simple life-events ontology in SU(M)O-KIF. *Proceedings of the Third International Conference EGOV 2002*, Springer, str. 112–119.
- [13] Berners-Lee, T. (2008). Sir Tim Berners-Lee Talks About the Semantic Web. *Transcript of the Podcast Interview with Paul Miller*. Najdeno 15. 7. 2008 na spletu: [http://talis-podcasts.s3.amazonaws.com/twt20080207\\_TimBL.html](http://talis-podcasts.s3.amazonaws.com/twt20080207_TimBL.html).
- [14] Blazquez, M., Fernandez, M., Garcia-Pinar, J., M., Gomez-Perez, A. (1998). Building Ontologies at the Knowledge Level using the Ontology Design

- Environment. *Proceedings of KAW'98 Eleventh Workshop on Knowledge Acquisition, Modeling and Management*.
- [15] Borgida, A., Brachman, R.J., McGuinness, L. D. in Resnick, A. L. (1989). CLASSIC: A Structural Data Model for Objects. *SIGMOD Conference*, str. 58–67.
  - [16] Brachman, R. (1977). *A Structural Paradigm for Representing Knowledge*. PhD Thesis. Harvard University.
  - [17] Brachman, R., Fikes, R. in Levesque, H. (1983). KRYPTON: a functional approach to knowledge representation. *IEEE Computer* 16 (1983) 67–73.
  - [18] Brickley, D. in Guha, R.V. (1999). *Resource Description Framework (RDF) Schema Specification*, W3C Proposed Recommendation.
  - [19] Brown, M.M. (2003). Electronic Government. V Rabin, J. (ed.). *Encyclopedia of Public Administration and Public Policy*, Marcel Dekker, str.427–432.
  - [20] Chen, P.S. (1976). The entity-relation model - toward a unified view of data. *ACM Transactions on Database Systems*, vol. 1, str. 9–36.
  - [21] Chun, S.A. in Atluri, V. (2003). Ontology-based Workflow Change Management for Flexible eGovernment Service Delivery, *The proceedings of the Third National Conference on Digital Government (dg.o 2003)*, May 18-21, Boston, MA.
  - [22] Corcho, O. in Gomez-Perez, A. (2000). A Roadmap to Ontology Specification Languages. *EKA'00*.
  - [23] Crevier, D. (1993). *AI: The Tumultuous Search for Artificial Intelligence*, New York, NY: BasicBooks, ISBN 0-465-02997-3, pp. 44–46.
  - [24] Cukjati, D., Vintar, M., Todorovski, L., Leben, A., Kunstelj, M. (2008). Razvoj modela življenjskih situacij v okviru e-uprave. V: *Uporabna informatika*. ISSN 1318-1882. Let. 16, št. 1, str. 33–43.
  - [25] Cukjati, D., Todorovski, L. (2008). Integrating E-services in Public Administration: Analysis of EU Research Projects and Involvement of Participants from SEE Region. *Symposium proceedings of Second International Symposium on the Development of Public Administration in South East (SE) Europe*, 19-20 June 2008, Ljubljana, Slovenia. ISBN 978-961-262-004-2.
  - [26] Cukjati, D., Rakar, I. (2009 - sprejeto za objavo, revija izide januarja 2009). Preverjanje semantične smiselnosti postavljenih pogojev s pomočjo propozicijske logike. *Revija Uprava*, ISSN 1581-7555.
  - [27] Davis, R., Shrobe, H. in Szolovits, P. (1993). What is a Knowledge Representation? *AI Magazine*, 14(1):17–33.
  - [28] Demuth, B. in Hussmann, H. (1999). Using OCL Constraints for Relational Database Design. <<UML>>'99 - *The Unified Modeling Language*, Springer LNCS 1723, pp. 598–613.
  - [29] Duineveld, A.J., Stoter, R., Weiden, M.R., Kenepa, B., Benjamins, V.R. (2000). WonderTools?: a comparative study of ontological engineering tools, *International Journal of Human-Computer Studies*, 52(6), str. 1111–1133.

- [30] European Commission. (2003). The Role of eGovernment for Europe's Future, COM(2003) 567 final, Brussels.
- [31] Feier, C., Roman, D., Polleres, A., Domingue, J., Stollberg, M., Fensel, D. (2005). Towards intelligent web services: Web Service Modeling Ontology (WSMO). V *Proc. of the International Conference on Intelligent Computing (ICIC 2005)*, China.
- [32] Fensel, D. (2004). *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*, 2nd ed. Berlin. Springer.
- [33] Fensel, D., Horrocks, I., van Harmelen, F., Decker, S., Erdmann, M. in Klein, M. (2000). OIL in a nutshell. *Proceedings of the European Knowledge Acquisition Conference (EKAW-2000), Lecture Notes In Artificial Intelligence*. Springer-Verlag.
- [34] Fountain, J., E., Kelman, S. in Kaboolian, L. (1992). Service to the Citizen: The Use of 800 Numbers in Government. *Customer Service Excellence: Using Information Technologies to Improve Service Delivery in Government*, Kennedy School of Government.
- [35] Garcia, R. (2008). *A Semantic Web Approach to Digital Rights Management*. PhD Thesis in Computer Science and Digital Communication, Universitat Pompeu Fabra, Barcelona, Spain. Najdeno na spletu 15. 12. 2008: <http://rhizomik.net/~roberto/thesis/>
- [36] Gartner. (2007). *Preparation for Update European Interoperability Framework 2.0 – - FINAL REPORT*. Gartner, 04-06 2007, Engagement: 221402470.
- [37] Genesereth, M. (1991). Knowledge Interchange Format. *Proceedings of the Second International Conference on the Principles of Knowledge Representation and Reasoning*. Morgan Kaufman Publishers, pp. 238–249.
- [38] Georgiadis, P., Lepouras, G., Vassilakis, C., Boukis, G., Tambouris, E., Gorilas, S., Davenport, E., Macintosh, A., Fraser, J. and Lockhead, D. (2002). A Governmental Knowledge-based Platform for Public Sector Online Services. *Proceedings of the 1st International Conference on Electronic Government-EGOV 2002*, pp. 362–369.
- [39] Gillies, J. in Cailliau, R. (2000). How the Web was Born: The Story of the World Wide Web. *Oxford University Press*. ISBN 0-19-286207-3
- [40] Gomez, P.A., Juristo, N., Pazos, J. (1995). Evaluation and assessment of knowledge sharing technology. V Mars, N.J. (ur.), *Towards Very Large Knowledge Bases. KBKS 95*. IOS Press, Amsterdam, str. 289–296
- [41] Gomez, P.A., Fernandez, L.M in Corcho, O. (2004). *Ontological Engineering with examples from the areas of Knowledge Management, e-commerce and the Semantic Web*. ISBN: 1-85233-551-3.
- [42] Gruber., R.T. (1993). A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5(2): 199–220.
- [43] Gugliotta, A., Roberto, V., Rowlatt, M., Davies, R., Cabral L., Domingue, J. (2005): A Semantic web service-based architecture for the interoperability of e-government services. V: *Web Information Systems Modeling Workshop (WISM 2005)*, in

- conjunction with The 5th International Conference on Web Engineering (ICWE 2005), Sydney, Australia.
- [44] Guha, R.V. in Lenat, D.B. (1989). *CycL: The Cyc representation language*. Technical Report ACT-CYC-452-89, MCC.
  - [45] Guijarro, L. (2007). Interoperability frameworks and enterprise architectures in e-government initiatives in Europe and the United States. *Government Information Quarterly*, 24(1), str. 90.
  - [46] Haarslev, V. in Möller, R. (2001). *RACER User's Guide and Reference Manual* Version 1.6. Technical report, University of Hamburg.
  - [47] Heflin, J. in Hendler, J. (2000). Searching the Web with SHOE. *AAAI-2000 Workshop on AI for Web Search*.
  - [48] Horrocks, I., Patel-Schneider, P., F. (2003). Reducing owl entailment to description logic satisfiability. V: *Proc. of the 2nd International Semantic Web Conference (ISWC)*.
  - [49] Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosz, B., Dean, M. (2004). *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*. Acknowledged W3C Member Submission. Najdeno 5. 12. 2008 na spletu: <http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/>.
  - [50] Horridge, M., Knublauch, M., H., Rector, A., Stevens, R. in Wroe, C. (2004). *A practical guide to building OWL ontologies using the Protege-OWL plugin and CO-ODE tools*. University of Manchester, CO-ODE project
  - [51] IDABC. *Interoperability*. Najdeno 1. 12. 2008 na spletu : <http://ec.europa.eu/idabc/en/chapter/5883>.
  - [52] IDABC. (2004). *EIF and IDA eLink: advancing e-government interoperability at pan-European level*. IDA Report, 21.3.2004. Najdeno 1. 12. 2008 na spletu: <http://ec.europa.eu/idabc/en/document/2358>.
  - [53] ISO/IEC JTC1 SC36: *Information Technology for Learning, Education, and Training*. Najdeno na spletu: <http://jtc1sc36.org/doc/36N0646.pdf>.
  - [54] Janssen, M., Hjort-Madsen, K. (2007). Analyzing Enterprise Architecture in National Governments: The Cases of Denmark and the Netherlands. *40th Annual Hawaii International Conference on System Sciences (HICSS'07)*, str. 218.
  - [55] Karp, D. P., Chaudhri, K.V. in Thomere, J. (1999). *XOL Specification - "an OKBC-friendly, XML-based ontology exchange language"*. version 0.3.
  - [56] Kifer, M., Lausen, G. in Wu, J. (1995). Logical foundations of object-oriented and frame-based languages. *Journal of the ACM (JACM)*, 42(4), str.741–843.
  - [57] Kodra, M., Patekar, S. (2008). *Ocena administrativnih stroškov – Poenostavitev postopka ustanovitve, spremembe ali izbrisa podjetja (enostavne družbe z omejeno odgovornostjo) z vzpostavitvijo informacijskega sistema e-VEM za g.d. MJU (Ministrstvo za javno upravo)*.

- [58] Knublauch, H., Fergerson, R.W., Noy, N.F. in Musen, M.A. (2004). The Protégé OWL Plugin: An Open Development Environment for Semantic Web applications. *Third International Semantic Web Conference*, Hiroshima, Japan.
- [59] Konstantas, D., Bourrières, J.P., Léonard, M., Boudjlida, N. (2006). *Interoperability of Enterprise Software and Applications*, 2006, Springer London, ISBN 978-1-84628-151-8
- [60] Kubicek, H. in Hagen, M. (2000). One-Stop-Government in Europe: An Overview. V: Kubicek, H. in Hagen, M. (eds): *One-Stop-Government in Europe. Results from 11 national surveys*. University of Bremen, str. 1–36
- [61] Laclavik, M., Zoltan, B.Z., Giang, N.T., Gatia, E., Hluchy, L. (2005). Methods for Presenting Ontological Knowledge to the Users. V: Popelinsky, L. in Kratky, M. (Eds.): *Znalosti 2005, Proceedings*, str. 61–64. ISBN 80-248-0755-6.
- [62] Lambrix, P., Habbouche, M., Perez, M. (2003). Evaluation of ontology development tools for bioinformatics. *Bioinformatics* 19, str. 1564–1571.
- [63] Lassila, O. in McGuinness, D.L. (2001). *The Role of Frame-Based Representation on the Semantic Web*. Knowledge Systems Laboratory Report KSL-01-02, Stanford University.
- [64] Leben, A., Kunstelj, M. (2004). E-portali in življenjske situacije. V: Vintar, M., Grad, J. (ur.). *E-uprava : izbrane razvojne perspektive*, (Upravna misel, 1, 04). Fakulteta za upravo, str. 81–93.
- [65] Leben, A., Kunstelj, M., Bohanec, M. (2004). Evaluation of life-event portal: a knowledge-based approach. V: *Proceedings of the Fourth European Conference on e-Government*. Dublin: ECEG, str. 496–480.
- [66] Luck, K., Nebel, B., Peltason, C. in Schmiedel, A. (1987) *The anatomy of the BACK system*. Technical Report KIT 41, Department of Computer Science, Technische Universitaet Berlin.
- [67] MacGregor, R., in Bates, R. (1987). *The Loom Knowledge Representation Language*. Technical Report, ISI/RS-87-188, USC/Information Sciences Institute.
- [68] McGuinness, D.L. in Harmelen, F. (2003). *OWL Web Ontology Language Overview*. W3C. Najdeno 15. 10. 2008 na spletu: <http://www.w3.org/TR/owl-features/>.
- [69] Minsky, M. (1974). *A Framework for Representing Knowledge*. Massachusetts Institute of Technology, Cambridge, MA.
- [70] Momotko, M., Izdebski, W., Tambouris, E., Tarabanis, K. in Vintar, M. (2007). An Architecture of Active Life Event Portals: Generic Workflow Approach. V: *Proceedings of the Sixth International EGOV Conference 2007*.
- [71] Motta, E. (1996). *OCML Manual*. KMI, The OU, Milton Keynes.
- [72] Noy, N., F. in McGuinness, D., L. (2001). *Ontology Development 101: A Guide to Creating Your First Ontology*. Knowledge Systems Laboratory.

- [73] Palvia, S.C.J. in Sharma, S.S. (2007). E-Government and E-Governance: Definitions/Domain Framework and Status around the World. ICEG. Najdeno 1. 10. 2008 na spletu: [http://www.iceg.net/2007/books/1/1\\_369.pdf](http://www.iceg.net/2007/books/1/1_369.pdf).
- [74] Peristeras, V. in Tarabanis K. A. (2005). Providing Pan-European E-Government Services with the Use of Semantic Web Services Technologies: A Generic Process Model. V *Proceedings of the Fourth International Conference*. Springer-Verlag, 226–236.
- [75] Peristeras, V., Konstantinos A. Tarabanis, K.,A. (2006). The Connection, Communication, Consolidation, Collaboration Interoperability Framework (C4IF) For Information Systems Interoperability. *IBIS* 1(1): 61–72.
- [76] Poole, D., Mackworth, A. in Goebel, R. (1998). *Computational Intelligence - A Logical Approach*. Oxford University Press, New York
- [77] Ramez, E. in Navathe, S.B. (2007). *Fundamentals of Database System*, 5th edition, Boston: Pearson/Addison Wesley. ISBN 978-0-321-36957-4.
- [78] Reinermann, H., Lucke, J. (2000). *Portale in der öffentliche Verwaltung*. Forschungsinstitute für öffentliche Verwaltung, Speyer, 2000.
- [79] Robinson, A. (2003). The Origins of Writing. V *Communication in History: Technology, Culture, Society*. Allyn and Bacon.
- [80] Rumbaugh, J., Jacobson, I. in Booch, G. (1998). *The Unified Modeling Language Reference Manual*, Addison-Wesley Pub Co.
- [81] Russell, S.J. in Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, ISBN 0-13-790395-2.
- [82] SAGA – Standards and Architectures for e-government Applications, Version 2.0. (2003). *KBSt Publication Series*, ISSN 0179-7263 Volume 59. Najdeno 15. 12. 2008 na spletu: [http://www.bsi.de/english/topics/egov/download/5\\_SAGA2\\_en.pdf](http://www.bsi.de/english/topics/egov/download/5_SAGA2_en.pdf).
- [83] Schwartz, D. G. (2006). An Aristotelian View of Knowledge Management. Vschwartz, D.G. (ur.), *Encyclopedia of Knowledge Management*, Idea Group Publishing, 2006.
- [84] Schellog, A., Mans, D. (2004). Citizens preferences towards One-Stop Government. *5th Annual International Conference on Digital Government Research, DG.O*, Seattle, WA.
- [85] Seidle, F. L. (1995). Rethinking the delivery of public services to citizens. Montreal, Institute for Research on Public Policy (IRPP).
- [86] SmartGov Consortium. (2004). SmartGov Project Deliverable D13: Final Project Report. Najdeno 11. 11. 2008 na spletu: [http://smartgov.e-gov.gr/deliverables/D13/SmartGov\\_D13\\_UoA\\_v03.pdf](http://smartgov.e-gov.gr/deliverables/D13/SmartGov_D13_UoA_v03.pdf).
- [87] Smith, M. K., Welty, C., McGuinness, D. L. (2004). *OWL Web Ontology Language Guide*. W3C. Najdeno 15. 10. 2008 na spletu: <http://www.w3.org/TR/owl-guide/>.
- [88] Sowa, J.F. (1976). Conceptual Graphs for a Data Base Interface. *IBM Journal of Research and Development*, 20(4), str. 336–357.



- [89] Sowa, J.F. (1987). Semantic Networks. *Encyclopedia of Artificial Intelligence*.
- [90] Studer, R., Benjamins, V.R. in Fensel, D. (1998). Knowledge engineering: principles and methods. *IEEE Transactions on Data and Knowledge Engineering*, 25(1–2), str. 161–197.
- [91] Šel, D., Gabrijel, T. (2007). Nacionalni interoperabilnostni okvir kot osnova za uspešno izvedbo povezovanja uradnih evidenc. *Predavanje na konferenci DSI 2007*.
- [92] Tambouris, E. (2001). An Integrated platform for Realising Online One-Stop Government: The eGov Project, V *Proceedings of the DEXA International Workshop "On the Way to Electronic Government"*, IEEE Computer Society Press, Los Alamitos, CA, str. 359-363, ISBN 0-7695-1230-5.
- [93] Tambouris, E., Tarabanis, K., Izdebski, W. in Momotko, M. (2006). Life-events revisited: Conceptualization and Representation using Generic Workflows. V *Workshop Proceedings of the 5th EGOV conference*, Universitätsverlag Rudolf Trauner.
- [94] Todorovski, L. (2007). *Life-event Analysis and Description Language*, Deliverable D12 in project OneStopGov. Najdeno 1. 12. 2008 na spletu: <http://www.onestopgov-project.org/index.php?name=UpDownload&req=getit&lid=459>
- [95] Todorovski, L., Leben A., Kunstelj, M., Cukjati, D., Vintar, M. (2006). Methodology for Building Models of Life Events for Active Portals. V: Gronlund, A. et al. (Eds.): *Communication proceedings of 5th EGOV International Conference*, EGOV 06, Trauner Verlag 2006, str. 61–68.
- [96] Tomažič, R., Krisper, M. (2001). Zasnova elektronskega poslovanja v javni upravi RS, *Uporabna informatika*, št. 4, str. 212–221.
- [97] United Nations. (2007). *e-Government Interoperability: A Review of Government Interoperability Frameworks in Selected Countries*. United Nations Development Programme, ISBN: 978-974-13-1624-3. Najdeno 16. 12. 2008 na spletu: <http://www.apdip.net/publications/iespprimers/eprimer-gif.pdf>.
- [98] Vintar, M. in Leben, A. (2002). The Concepts of an Active Life-Event Public Portal. V *Proceedings of the First International Conference EGOV 2002*. LNCS 2456, Springer, str. 383–390.
- [99] W3C. (2008). *OWL Working Group*. Najdeno 15. 7. 2008 na spletu: [http://www.w3.org/2007/OWL/wiki/OWL\\_Working\\_Group](http://www.w3.org/2007/OWL/wiki/OWL_Working_Group).
- [100] *Why OWL-RDFS has such poor web presence?* Najdeno 16. 12. 2008 na spletu: <http://ontologyonline.blogspot.com/2007/11/embedding-owl-rdfs-syntax-in-xhtml-with.html>.
- [101] Wimmer, M.A. in Tambouris, E. (2002). Online One-Stop Government: A working framework and requirements. V: Traunmuller, R. (Ed.) *Information Systems: The E-Business Challenge*, Kluwer Academic Publishers, str. 117–130.



## **PRILOGA**

Priloga A: Spisek pogosto uporabljenih kratic:

ŽS – Življenjska situacija

SJU – Storitve javne uprave

PZ – Predstavitev znanja

IKT – Informacijsko-komunikacijska tehnologija

UI – Umetna inteligenca

HKOM – Hitro komunikacijsko omrežje