



Nõuded RIHA ontoloogiatele

Versioon 2011/r2.6
21.10.2011

Hele-Mai Haav
Küberneetika Instituut

1. Sissejuhatus ja taust.

Käesolev dokument põhineb *Semantika juhisel*¹ täpsustades seda praktiliste nõuete ja juhistega, mis on vajalikud RIHAsse paigutatavate ontoloogiate kvaliteetseks loomiseks (vt ka RIHA määrus²).

See juhise on mõeldud kasutamiseks koos *Semantilise kirjeldamise juhise*³ ja *Ontoloogiate loomise metoodika*⁴ dokumendiga. Käesolev dokument vaatleb valdkonna ontoloogiate loomist ja esitab nõuded ontoloogiate vormistamiseks. *Semantilise kirjeldamise juhise* käsitleb andmeobjektide ning andmeteente semantiliste kirjelduste (edaspidi semantilised kirjeldused) koostamist kasutades loodud ontoloogiaid. *Ontoloogiate loomise metoodika* dokumendis esitatakse erinevad võimalikud metoodilised stsenaariumid nõuetele vastavate ontoloogiate ja semantiliste kirjelduste loomiseks.

Käesolev dokument on mõeldud kasutamiseks valdkonnaekspertidele, analüütikutele, arendajatele ja teistele semantikavarade loojatele.

Ontoloogiate loomine on illustreeritud eluliste näidetega riigi infosüsteemide (IS) valdkonnast.

2. Ontoloogiate kasutusala riigi infosüsteemis.

2.1. Ontoloogiate osa infosüsteemi arenduses.

RIHAsse paigaldatavad ontoloogiad on mõeldud tagamaks riigi infosüsteemi (IS) semantilist koosvõimet andmekogude andmeväljade ja andmeteente semantilise kirjeldamise kaudu. Viimane ei tohiks olla siiski ainus ontoloogiate rakendusvaldkond ja seega oleks hea valdkonna ontoloogiad luua taaskasutatavad.

Ontoloogiate arenduse sidumine vastava IS arendustsükliga jääb praegu iga IS valdaja (tellija) otsustada. Sisuliselt võiks ontoloogia tegemise nõude lisada IS hankedokumentidesse. Kui arendaja teeb IS analüüsi, siis on parem kui ta ise ka tegeleks vastava ontoloogia arendusega. Et analüüsifaasis osaleb ka tellija, siis on just ontoloogia kontseptualiseerimise ajal tellijapoolse eksperdi valdkonnaalased teadmised väga vajalikud. Seega me soovime ontoloogia arenduse läbi viia tellija ja arendaja koostöös. See annab lootuse, et ontoloogia kajastab valdkonna teadmisi. Teisalt on ontoloogia väga hea vahend arendajaga ISis

¹ http://www.riso.ee/et/files/Semantika_juhis_v1.1.pdf

² <https://www.riigiteataja.ee/ert/act.jsp?id=12933746&searchCurrent>

³ P. Kungas, *Semantilise kirjeldamise juhise*, Tallinn, 2010

⁴ H-M. Haav, *Ontoloogiate loomise metoodika*, Tallinn, 2010

kajastatavate andmete üle arutlemiseks. Ontoloogia realiseerimine (näiteks kodeerimine OWLi) ja selle abil andmeteenuste semantiline kirjeldamine võiks jääda põhiliselt arendaja tööks. Aga on väga hea kui tellija teab, mida ta tellib.

2.2. RIHA kui valdkonna ontoloogiate hoidla ja otsingusüsteem.

RIHA kogub ontoloogiaid ja nende abil loodud andmeväljade ning andmeteenuste semantilisi kirjeldusi. Peale selle toimib RIHA kui vastavate semantikavarade otsingumootor. Üheks RIHA tulevikusuunaks on võimaldada ontoloogia loojate omavahelist suhtlust vastava foorumi abil selleks, et vahetada teadmisi ja arutada praktilisi ontoloogiate loomise probleeme. RIHA võimaldab tulevikus ontoloogiapõhist ISide koosvõime analüüsi ja andmeteenuste otsingut. See on abiks uute ISide ja andmeteenuste loomisel.

3. Valdonnaontoloogiate skoop RIHAs.

3.1. Valdcondade skoop.

RIHAs esitatud valdkonnad on võetud järgmisest indeksist: http://europa.eu/pol/index_et.htm. Vastavalt valdkonna ekspertide nõudmisele on võimalik nende väga üldiste valdkondade alla tekitada täpsustavaid alamvaldkondi. Praegu on RIHAs piirang, et ühe valdkonna alla saab üles laadida täpselt ühe ontoloogia. Seega tuleb iga ontoloogia jaoks tekitada uus alamvaldkond sobiva valdkonna alla.

3.2. Ontoloogiate eesmärk ja skoop.

Valdkonna ontoloogiate loomise *eesmärk* on tagada riigi ISde semantiline koosvõime võimaldades nii andmekogude andmeobjektide kui andmeteenuste kirjelduste semantilist rikastamist vastava valdkonna ontoloogia mõistetega.

Valdkonna ontoloogia *skoobi* määravad mitmed asjaolud. Põhilised skoopi mõjutavad faktorid sõltuvad järgmistest ISi olekutest:

1. Olemasolev IS ilma andmeteenusteta
2. Olemasolev IS koos andmeteenustega
3. Uus loodav IS

3.2.1. Olemasolev IS ilma andmeteenusteta.

Valdkonna ontoloogia hõivab sel juhul minimaalselt kõik andmekogu põhiandmete väljad jättes välja andmebaasi tabelite võtmeväljad. Reeglina on andmekogu andmeväljade ontoloogialiseks vasteks andmetüübiomadus (owl:DatatypeProperty). Valdonna mudel on sisuliselt laiem kui andmebaasi mudel, seega valdkonna ontoloogia võib hõivata lisaks andmebaasi tabelitele ja andmeväljadele vastavatele mõistetele, seostele ning omadustele ka muid valdkonnast arusaamiseks vajalikke komponente.

3.2.2. Olemasolev IS koos andmeteenustega

Lisaks ülaltoodud skoobile peab valdkonna ontoloogia hõivama kõiki andmeteenuste sisend/väljundparameetritega kaetud andmevälju. Seega kui andmeteenuste parameetritena esinevad andmekogu lisaandmete väljad, mis pole veel kaetud mingi teise valdkonna ontoloogiaga, siis tuleb ka nendele vastavad mõisted lülitada valdkonna ontoloogiasse. Ontoloogialises mõttes piisab lisaandmete korral vastavate klasside esitamisest, kuid kommentaaride osas tuleb selgitada mõiste sisu ja kasutusjuhtum andmeteenustes (vt täpsustamiseks punkt 4.2).

3.2.3. Uus loodav IS

Uue IS korral võetakse ontoloogia loomise aluseks IS kontseptuaalne mudel ja seega kajastab ontoloogia antud juhul põhiandmete välju. Kui uues ISis planeeritakse ka uusi andmeteenuseid, siis on vaja jälgida et andmeteenuste sisend/väljund oleks kaetud ontoloogia mõistetega. Näiteks võib vaja minna mõnda lisaandmevälja, mida pole veel vastava valdkonna ontoloogias olemas jms. Põhimõtteliselt võib uue IS korral alustada valdkonna mudeli loomisest ontoloogia kujul ja siis selle alusel teha IS andmekogu kontseptuaalse mudeli. Vastava valiku määravad ära konkreetse IS loomise tingimused.

Kõigi valdkondade ontoloogiatega skoobist jäävad välja valdkonnas kasutatavad klassifikaatorid. Viimaste korral ontoloogiseeritakse ainult nende metaandmed ja vastav ontoloogia saab olema RIHA ontoloogia osa.

4. Nõuded ontoloogiatele.

Vastavalt Semantika juhisele esitatakse valdkonna ontoloogiad OWL keeles ja paigutatakse RIHAsse. RIHA eksperdid kontrollivad ontoloogiatega ja semantiliste kirjeldustega vastavust nõuetele enne nende lõplikku publitseerimist.

4.1. Ontoloogiatega vormistamine.

4.1.1. Ontoloogia unikaalne aadress.

Valdkonna ontoloogia aadress veebis (st URI või IRI) RIHAs on järgmisel kujul: <https://riha.eesti.ee/riha/onto/valdkond.alamvaldkond/aasta/versiooninumber>. Aasta esitatakse kujul yyyy, näiteks 2010. Versiooni numbritena kasutatakse notatsiooni "r1","r2" jne näitamaks mitmendat konkreetset aastal väljalastud reliaasi silmas peetakse.

Ontoloogia URI on vajalik ontoloogia OWL kirjelduse komponentide nimeruumi määramiseks. Nimeruume vajatakse ontoloogia kirjelduses selleks, et täpselt näidata, milliseid termineid millistest erinevatest sõnastikest on kasutatud ontoloogia kirjeldamisel. OWLis esitatud ontoloogiad kasutavad reeglina peale OWLi enda ka RDF, RDFS ja xsd sõnastikke.

Näiteks vaatleme fragmenti Äriregistri ontoloogia URI kirjeldusest RDF/XML vormingus.

```
<!DOCTYPE rdf:RDF [
  <!ENTITY r1
"https://riha.eesti.ee/riha/onto/ettevotlus.ariregister/2010/r1/">... ]
  <rdf:RDF
xmlns="https://riha.eesti.ee/riha/onto/ettevotlus.ariregister/2010/r1#"
  xml:base="https://riha.eesti.ee/riha/onto/ettevotlus.ariregister/2010/r
1
xmlns:r1=
"https://riha.eesti.ee/riha/onto/ettevotlus.ariregister/2010/r1.owl">
```

Selleks, et ontoloogiatega RDF/XML vorming oleks eksponeeritav kõigis veebilehitsejates, peaks standardse lühinime eraldava sümboli „#“ asemel kasutama sümbolit „/“, mis on ka lubatud. Ontoloogiarektor Protégé võimaldab uue ontoloogia loomisel määrata, milliseid eraldajaid soovitakse ontoloogia komponentide URI (või IRI) esitamisel kasutada. Ülaltoodud näites on kasutatud eraldajat „/“. Ontoloogia komponendile viidatakse OWL kirjelduse RDF/XML vormingus sel juhul kasutades lühikuju **"/komponendi nimi"/**. Näiteks, lühinimi **"/Notar"/** viitab klassile Notar käesolevast ontoloogiast.

4.1.2. Ontoloogia selgitused.

Ontoloogiate selgitused on soovitatav esitada mitmes keeles, kuigi see pole praegu otsene nõue. Mitmekeelsed selgitused võimaldavad ontoloogiate taaskasutust rahvusvahelisel tasemel ja soodustavad meie IS koosvõimet teiste riikide infosüsteemidega. Lisaks eesti keelele võiks selgitused olla näiteks ka inglise keeles.

Ontoloogia selgitused võib jaotada kahte gruppi: ontoloogia kui terviku selgitused ja ontoloogia komponentide selgitused.

Ontoloogia kui terviku kohta esitatakse ka versiooni info märgendi **owl:versionInfo** abil.

Ontoloogia ja tema komponentide selgituste esitamiseks on RIHAsse paigutatavate ontoloogiate korral kohustuslikud järgmised OWLs kasutatavad märgendid (vt ka OWL keele spetsifikatsiooni⁵):

rdfs:label - ontoloogia või selle komponendi inimesele esitamisele orienteeritud nimi. Neid nimesid võib olla mitu ja nad seotakse alati konkreetse inimkeelega (atribuudi **xml:lang** abil, eesti keelt tähistab „et“ ja inglise keelt „en“ vastavas standardis ISO 639-1).

rdfs:comment – ontoloogia või selle komponendi definitsioon (selgitus) mingis inimkeeles. Selle võib esitada ka mitmes keeles.

Ontoloogiate OWL kirjelduses esinevad atribuudid **rdf:about** ja **rdf:ID** on mõeldud ontoloogia või ontoloogia komponendile viitamiseks ning nad on orienteeritud masintöötlemisele mitte inimesele.

Vaatame järgmist Äriregistri ontoloogia kui terviku selgitust RDF/XML vormingus:

```
<owl:Ontology rdf:about="">
  <rdfs:label xml:lang="et"
    >#196;rregistri ontoloogia</rdfs:label>
  <rdfs:comment xml:lang="et"
    >Eesti #196;rregistri ontoloogia, mis h#245;lmab
    #196;rregistri X-tee teenuste sisend- ja
    v#228;ljundandmeid.</rdfs:comment>
  <owl:versionInfo xml:lang="et">ver1</owl:versionInfo>
</owl:Ontology>
```

Selles näites antakse Äriregistri ontoloogia eestikeelne selgitus. Atribuut **rdf:about=""** tähendab, et ontoloogia masintöödeldavaks viiteks/nimeks on vaikimisi tema URI. Märgend **rdfs:label** koos **xml:lang** atribuudiga “et” ütleb, et tegemist on eestikeelse nimetusega, milleks on Äriregistri ontoloogia (täpitähed on kodeeritud). Järgmises näites on toodud selle ontoloogia klassi *Registriandmed* selgitus eesti keeles. Klassi masintöödeldav nimi on antud atribuudiga **rdf:about="/Registriandmed"/**, mis näitab, et nimi *Registriandmed* identifitseerib vastava klassi vaadeldava ontoloogia sees.

```
<owl:Class rdf:about="/Registriandmed"/>
  <rdfs:label xml:lang="et">Registriandmed</rdfs:label>
  <rdfs:subClassOf rdf:resource="&owl;Thing"/>
  <rdfs:comment xml:lang="et"
    >K#245;ik registripidaja poolt registreeritud isiku kohta
    k#228;ivad andmed (nt #228;rinimi, majandusaasta periood)</rdfs:comment>
</owl:Class>
```

RIHA valdkonna sõnastiku terminite loetelu kuvamisel näidatakse ontoloogia komponendi nimetus ja määratlus vastavalt ontoloogia OWL kirjelduses esinevatele elementidele **rdfs:label** ja **rdfs:comment**.

⁵ <http://www.w3.org/TR/2004/REC-owl-guide-20040210/#SimpleClasses>

Ülalpool esitatud ontoloogia kirjelduste vormistamisel on abiks ontoloogiaredaktorid (näiteks Protégé), mis võimaldavad hõlpsasti tekitada ontoloogia OWL kirjeldust kaasa arvatud nimetatud selgituste osad. Selgituste osa nimetatakse seal üldistatult annotatsiooniks.

4.1.3. Nimetamine.

Klasside, seoste ja omaduste nimetamise reeglid on toodud kehtivas Semantika juhises punkti 3 all. Samas on ka juhised kommentaaridele.

4.1.4. Andmetüübiomaduste ulatuse määramine.

Andmetüübi ulatuses (muutumispiirkonnas) tohib 2004 aasta OWLi järgi kasutada XSD baastüüpe (RDF literal'i kasutamist, mida OWLis samuti saab andmetüübiomaduste ulatuse määramisel kasutada, ei soovitata). Juhul kui ei osata teatud elemendi sisu täpsemalt määrata võiks kasutada anySimpleType otseseid alamtüüpe ("**Primitive** datatypes") lingilt <http://www.w3.org/TR/xmlschema-2/> (punktis "3.2 Primitive datatypes" on primitiivsete andmetüüpide sisu avatud).

4.2. Andmekogu lisaandmete esitamine ontoloogias.

Üks üldisem probleem riigi ISidele vastavate valdkondade ontoloogiseerimise juures on see, et iga IS kasutab peale oma põhiantmete (mõistete) ka andmeid teistest ISidest. Kui meil pole veel käepärast nn võõr-ISi ontoloogiat, siis ei saa ka selle mõisteid taaskasutada ega neile semantilisel kirjeldamisel viidata. Seoses sellega keskendutakse praegu ainult iga ISiga hõivatud valdkonna põhiantmete ontoloogiseerimisele. See kitsendab ontoloogiate skoopi ja tagab pädevate valdkonna ekspertide kasutamise.

Andmeteenuseid omavate olemasolevate ISide korral (vt punkt 3.2.2) tekib probleem nende andmeteenuste kirjelduste semantilisel kirjeldamisel, mille sisend/väljund parameetrid vastavad mõne lisaandmetega hõivatud mõistele, sest pole võtta vastavaid mõiste definitsioone. Selle probleemi saab lahendada luues valdkonna ontoloogiasse klassi nimetusega *Lisandmed* ja andmekogude lisaandmeid kirjeldavad klassid kirjeldada eraldi selle klassi alamklassidena nn kohahoidjatena, et saaks pärast sujuvalt vastava valdkonna ontoloogia kasutusele võtta. Lisaandmete osas piisab teadmiste kirjeldamisest klasside tasemel. Hiljem kui lisaandmetele vastavad ontoloogiad on õiges kohas korrektselt kirjeldatud, siis saab selle lisaandmete klassi koos alamklassidega oma IS ontoloogiast eemaldada kui mittevajaliku.

4.3. Valdkonna ontoloogia seos teiste ontoloogiatega.

Valdkonnad võivad omada ühiseid mõisteid ja seega ei ole ka valdkonna ontoloogiad isoleeritud vaid seotud teiste valdkondade ontoloogiatega. Valdkondadele ühised mõisted tuleks mitmepoolselt kooskõlastada vastavate ekspertide poolt. Teiste valdkondade või mujal loodud ontoloogiate täielik või osaline kasutamine mingi ontoloogia loomisel on tavapärase tegevus, sest ontoloogiad ongi mõeldud valdkonna teadmiste taaskasutamiseks.

Ontoloogiate kirjeldamisel OWL keeles on mitmeid mooduseid kuidas mingi valdkonna ontoloogias kasutada mõne teise ontoloogia mõisteid. Me vaatleme neist järgmisi:

- Ontoloogia import
- Viitamine teiste ontoloogiate komponentidele.

Ontoloogia import tähendab, et aktsepteeritakse täielikult mingi teise valdkonna ontoloogiat loodava ontoloogia osana. Märge owl:imports abil näidatakse millist ontoloogiat milliselt URIlt imporditakse. Importimisel on tähtis mees pidada, et imporditavat ontoloogiat haldab ikkagi tema arendaja ning iga kord kui pöördutakse kasutamiseks loodava ontoloogia poole, imporditakse jooksvalt loodavasse ontoloogiasse ka imporditav ontoloogia vastavalt URIlt. Imporditav ontoloogia võib omakorda sisaldada imporditud ontoloogiad jne. Võib juhtuda, et imporditav ontoloogia on vahepeal muutunud näidatud viitelt kättesaamatuks. Kui imporditava ontoloogia arendaja oli vahepeal teinud muutusi, siis ilmnevad need ka loodud ontoloogias.

Teine moodus on kasutada nimeruumide deklareerimist selleks, et viidata nimedele, mis on defineeritud teistes OWLis kirjeldatud ontoloogiates. Ontoloogia OWL kirjelduses on võimalik viidata mingi teise ontoloogia komponendile. Selleks tuleb ontoloogia komponendi masinloetav nimi (identifikaator) panna viitama mõne sobiva ontoloogia vastavale mõistele või mõnele teisele komponendile. Seega aktsepteeritakse oma ontoloogia komponendina mõne teise ontoloogia komponenti koos tema definitsiooniga. Sel juhul peab enne jälgima, et vastav komponent ikka tõesti sobib oma definitsiooni poolest loodavasse ontoloogiasse.

Näiteks võime lisada Äriregistri ontoloogiasse uue nimeruumi, mis vastab e-notari ontoloogiale:

```
xmlns:v13=https://riha.eesti.ee/riha/onto/oigus/e_notar/2010/r1
```

Oletame, et meil on vaja Äriregistri ontoloogias viidata Notari mõistele, mis on defineeritud e-notari ontoloogias, siis saame luua klassi, mille masintöödeldav nimi on järgmine:

```
<owl:Class rdf:about="&v13;Notar">
```

Lühikuju "&v13;Notar", kus v13 tähistab vastavat ontoloogiat ja Notar on selle ontoloogia komponendi nimi on veel üks ontoloogia komponendile viitamise võimalus. Ontoloogiarektor Protégé võimaldab sooritada mõlemaid tegevusi nii importi kui viitamist.

5. Valdkonna teadmiste kontseptualiseerimise probleemid.

Valdkonna teadmiste kontseptualiseerimisel ja ontoloogiatesse sättimisel võib tekkida vigu. Allpool vaatleme mõningaid tüüpilisi vigu.

- Ontoloogiate alt-üles loomisel tekib tüüpiliselt palju liiasust nii atribuutide kui omaduste osas. Liiasuse vältimiseks piisab kui defineerida teatud ühised omadused ülemklassis ning need päritakse taksonoomilise pärimissuhte alusel alamklassis.
- Tihti leiame taksonoomias mõisteid, millel on ainult üks alamõiste. Näiteks järgmises taksonoomias asi->kinnisasi->kinnistu on just selline situatsioon. Kui pole ette näha, et lisandub uusi alamõisteid, siis tuleks loobuda sellisest klassifikatsioonist ja tuua alamõisted ülemisele tasemele. Siiski tuleb alati silmast pidada, et ülemklass/alamklass seos on taksonoomiline ja peab saama ütelda, et iga alamklassi individid on ka tema ülemklassi individid selle ontoloogia piires.
- Tihti tekib ontoloogiates loogikavigu mõne seose ulatuse (muutumispiirkonna) määramisel. Näiteks, soovime kirjeldada, et klassi Dokument individid on seotud seose dok_sisaldab_parandajad_ ja_parijad abil klasside Parandajad ja Parijad individidega. Sel juhul on selle seose ulatuseks Parandajad OR Parijad aga mitte Parandajad AND Parijad. Viimane esitab nimetatud klasside individide ühisosa aga mitte summat. See on väga tüüpiline loogikaviga. Kui kasutada järjest mitut domeeni

(või ulatuse) klassi, siis see tähendab, et seose (ObjectProperty) domeeniks või ulatuseks võetakse näidatud klasside ühisosa (intersection).

- Üldjuhul tuleb OWLis kirjeldatud ontoloogiates määrata, et klassid on vastastikku lõikumatud (Disjoint). See garanteerib, et mingi klassi indiviidid ei saa olla samaaegselt mingi teise klassi indiviidid. Vaikimisi on OWLis kõik klassid lõikumavad.
- Seosel või andmetüübiomadusel võib olla funktsionaalsuse omadus. Funktsionaalseks võib seose või atribuudi märkida ainult juhul kui ta seob ulatuse klassi indiviididest maksimaalselt 1 indiviidi (st kardinaalsus on 0..1). Reeglina on selliste omadustega atribuudid ehk andmetüübiomadused.

6. Ontoloogiate disaini mustrid.

Ontoloogiate arenduses hakkavad korduma teatud disaini mustrid, mida on kasulik teada ja taaskasutada. Erinevaid mustreid võib leida veebilehelt ontologydesignpatterns.org, kuhu ontoloogiate loojad on kogunud erinevaid korduvaid disaini mustreid nende taaskasutamise eesmärgil. Allpool on esitatud 3 meie riigi IS ontoloogiate loomisel tihti vajalikku disaini mustrit, milleks on inspiratsiooni saanud nimetatud veebilehelt.

6.1. Omaduste puntrad.

Omaduste/seoste pundar on suhteliselt suur hulk ühiseid omadusi/seoseid, mida omavad mingi hulk klasse. (NB! mitte segi ajada ülemklass/alamklass seosega ja sellega seonduva omaduste päritavusega). Selleks, et taolised omaduste puntrad ei halvendaks ontoloogia ülesehitust tuleks nad koondada mingi uue klassi alla ja luua ka vastav seos, mis seoks selle uue klassi ja omaduste puntrasse kuuluvad omadused. Originaalsed klassid tuleks sel juhul siduda selle uue klassiga selle asemel, et siduda neid kõiki eraldi iga puntrasse kuuluva omadusega.

Olgu näiteks klassidel $\{Firma, Inimene\}$ ühiste omaduste (owl:DatatypeProperty) pundar $\{nimi, aadress, telefoninumber\}$. Selle korrastamiseks tuleks ontoloogiasse luua uus klass, näiteks *Kontaktandmed*, mille indiviidide atribuutideks oleksid siis sellesse puntrasse kuuluvad omadused. Klassid *Firma* ja *Inimene* tuleb näiteks seose *omab_kontaktandmeid* läbi siduda klassiga *Kontaktandmed*, kusjuures selle seose domeeniks on klasside summa *Firma OR Inimene* ja ulatuseks on klass *Kontaktandmed*. Samasugust meetodit saab kasutada ka seoste (owl:ObjectProperty) pundarde korral.

6.2. Binaarne seos atribuudiga.

Üldiselt saab OWLs väljendada ainult binaarseid seoseid so seoseid 2 klassi indiviidide vahel. Seda tehakse konstruktsiooni owl:ObjectProperty abil. Seosed ise ei oma atribuute. Juhul kui soovitakse modelleerida situatsiooni, kus seosel endal on ka atribuut või atribuudid, siis tuleb luua uus klass sellise seose esitamiseks. Näiteks, olgu meil vaja väljendada seost kahe asukoha vahel ja samal ajal näidata nende asukohtade vahelist kaugust kilomeetrites. Kontseptuaalselt on kaugus sel juhul seose atribuut.

Ontoloogias esitatakse see järgmiselt. Kõigepealt luuakse klassid *Asukoht* ja *Kauguseseos*. Klassi *Kauguseseos* indiviididega on seotud atribuut (owl:DatatypeProperty) *Kaugus*, mille tüüp on näiteks *xsd.float*. Järgmisena luuakse seos (owl:ObjectProperty) *seob_punkte*, mille domeen on klass *Kauguseseos* ja ulatus on klass *Asukoht*.

Klassi *Kauguseseos* defineerime järgmiselt:

```
Kaugus exactly 1 float
and seob_punkte exactly 2 Asukoht
```

Võib juhtuda, et on vaja kasutada keerulisemaid mitmeseid seoseid. Sel juhul on abiks järgmine link <http://www.w3.org/TR/swbp-n-aryRelations/>.

6.3. Rollid.

Tihti on vaja ontoloogias modelleerida rolle. Rollide kujutamine ontoloogiates ei ole lihtne, sest rollid on midagi, mis sõltuvad kontekstist so mingite sündmuste juhtumisest. Näiteks inimene jääb inimeseks sõltumatult sellest, et ta on töötaja rollis, lapsevanema rollis, maksumaksja rollis jms. Seega peaksid rollid olema seotud nii olemitega kui sündmustega. Ontoloogiate arenduses pole praegu head ühtset käsitlust rollide modelleerimiseks.

Lihtsamad piiratud käsitlused kujutavad erinevaid rolle kas klasside taksonoomiatena või üldise klassi Roll indiviididena. Rolle kandvad subjektid/objektid on modelleeritud klasside abil. Mõlemal juhul saab vastavaid rolle siduda vastavate subjektidega/objektidega läbi seose. Näiteks, olgu meil subjektide klasside hulgas klass Isik ja rollide klasside hulgas klass Notar, siis saame tekitada seose Isik <omab_rolli> Notar.

7. Ontoloogiate loomise vahendid.

Ontoloogiate loomiseks on võimalik kasutada suhteliselt kasutajasõbralikke tarkvaralisi vahendeid. Käesolevas punktis antakse soovitusel nimetatud vahendite kasutamiseks ja paigaldamiseks. Me vaatleme põhiliselt vabavara.

7.1. Ontoogiaredaktor Protégé.

Ontoogiaredaktor Protégé on maailmas enimkasutatav ontoogiaredaktor. Ta on loodud Stanfordini Ülikooli biomeditsiinikeskuses USA-s. Tarkvara saab alla laadida järgmiselt veebilehelt: <http://protege.stanford.edu/download/registered.html#p4>. Seal on ka kasutusjuhend ja muu info redaktori ning ontoologiade loomise kohta. Ontoogiaredaktor Protégé kasutab ontoologiade visualiseerimiseks põhiliselt programmi Graphviz. Selle programmi saab alla laadida järgmiselt veebilehelt: http://www.graphviz.org/Download_windows.php. Kui kasutatakse MS Vista operatsioonisüsteemi, siis tuleb tarkvara paigaldada käsurealt

`msiexec /a graphviz-x.xx.msi`.

Selleks, et Protégé ja Graphviz koos toimiksid läbitakse järgmised sammud:

- avatakse Protege, valitakse File menüüst Preferences ja sealt OWLViz tab;
- juhul kui on tegemist versiooniga Graphviz2.26.3 ja standardse paigaldusega, siis pannakse Dot Application Path lahtrisse C:\Program Files\Graphviz2.26.3\bin\dot.exe;
- juhul kui on paigaldatud mõni teine Graphviz versioon või programmi paigutus on mittestandardne, siis tuleb vastavalt sellele spetsifitseerida nii programmi nimi kui juurdepääsutee.

Ontoologiade dokumenteerimisel võib kasutada ka vahendit IHMC CmapTools COE v5.0, mis on mõistekaartidel põhinev ontoologiade visualiseerimise vahend. Programmi saab järgmiselt veebilehelt: http://cmapdownload.ihmc.us/coe/Web_InstallersV5.0/install.htm.

7.2. Vahendeid UML2OWL konverteerimiseks.

Paratamatult tekib mõte, et ontoologiaid võiks tekitada automaatselt või poolautomaatselt juba olemasolevatest andmekogu UML mudelitest või XMI failidest. Nii saaksime andmebaasi mudelile vastava ontoologia ehk andmeontoloogia. Teatavasti on andmebaasi mudel skoobilt kitsam kui valdkonna mudel, kuid mõningatel juhtudel või andmeontoloogia olla abiks valdkonnaontoloogia loomisel.

Sel eesmärgil on loodud ja loomisel vahendid ning keeled, mis võimaldavad transformeerida UML või XMI esitused OWL ontoologiaks. Nendesse võimalustesse tuleb

siiski kriitiliselt suhtuda, sest ei saa loota, et taolise konverteerimise teel saadud ontoloogia on täisväärtuslik valdkonna mudel või isegi andmebaasi mudel. Osa semantikat läheb paratamatult kaduma.

IBMi poolt esitati OMGle (Object Management Group) ontoloogiade defineerimise metamudel ODM (Ontology Definition Metamodel), mille abil saab UML mudelit teisendada OWL ontoloogiaks (vt <http://www.omg.org/spec/ODM/1.0/>). IBM on ka realiseerinud ODMi.

Laiemat kasutamist leiab Eclipse realiseeritud ODMi variant (vt lingilt <http://www.eclipse.org/m2m/atl/atlTransformations/#UML2OWL>).

Vahendid XMI2OWL konverteerimiseks on järgmised:

- CIMTOOL (<http://wiki.cimtool.org/Download.html>)
- Poseidon (<http://www.gentleware.com/index.php?id=15>)
- TopBraid Composer (http://www.topquadrant.com/products/TB_Composer.html)

8. Kokkuvõte.

Ontoloogiade loomine on keeruline ülesanne. Loodetavasti on käesolev praktiline juhendmaterjal abiks selle ülesande lahendamisel riigi ISi valdkondades.

Käesoleva juhise Lisas 1 on toodud kontrollküsimused selleks, et enne ontoloogia RIHAsse paigutamist saaks ontoloogia kvaliteeti nende abil hinnata ja esinevad puudused kõrvaldada.

Juhise LISA 2 esitab sõnastiku ontoloogiatehnika põhimõistest.

LISA 1.

Ontoloogia kirjelduse kvaliteedi hindamise leht

Enne kui ontoloogia RIHAsse paigutada tuleks kontrollida ontoloogia kirjelduse vastavust järgmistele põhinõuetele (vt tabel 1).

Tabel 1. Ontoloogia kirjelduse kvaliteedi hindamise kontrollküsimused.

Semantilise kirjeldamise aspekt	Kontrollküsimus
<i>Üldised ontoloogia omadused ja vormistus</i>	
Ontoloogia valdkond ja URI (või IRI)	Kas ontoloogia valdkond on RIHAs kooskõlastatud ja kas ontoloogia URI (või IRI) on vastav valdkonnale? Kas ontoloogia versioon on õige?
Ontoloogia annotatsioon	Kas ontoloogia kui terviku selgitused on toodud st rdfs:label ja rdfs:comment ja versiooni info owl:versionInfo omaduste väärtused ?
Ontoloogia komponentide annotatsioonid	Kas kõikide ontoloogia komponentide selgitused on esitatud st rdfs:label ja rdfs:comment omaduste väärtused?
Ontoloogia komponentide nimede eraldajad	Kas eraldajatena on kasutatud sümbolit „/“ aga mitte sümbolit „#“ ?
Ontoloogia komponentide nimede ja selgituste vastavus Semantika juhises esitatud nimetamise reeglitele	Kas ontoloogia komponentide nimed ja selgituste tekst vastavad Semantika juhises esitatud nimetamise reeglitele (näiteks täpitähtede ja suurtähtede jms kasutamise osas)?
Ontoloogia selgituste arusaadavus	Kas ontoloogias kasutatavad selgitused on piisavalt arusaadavalt esitatud ka valdkonna mitte-ekspertidele.
<i>OWL konstruktsioonid</i>	
URId/nimeruumid	Kas URId/nimeruumid vastavad nõuetele? (vt ka imporditud ontoloogiate osas)
Lõikumatus (Disjoint) määratlus klassidel	Kas kõik lõikumatud klassid on määratletud vastavalt?
Domeen (Domain), ulatus (Range)	Kas seostel on määratud domeen ja ulatus? Kas atribuutidel (omadustel) on määratud andmetüübi ulatus? Kui domeeni või ulatuse kirjeldamisel on kasutatud mitut klassi, siis kas nende klasside vahel on kasutatud sisuliselt õiget loogilist operaatorit (or/and)? (vt punkt 5)
<i>Taksonoomia ja disaini vead</i>	
Tsüklid	Kas klasside taksonoomias esineb tsükleid?

Alamklass/ülemklass	Kas klassi kõik sama taseme alamklassid on samal abstraktsiooni tasemel? Kas klassil on ainult üks alamklass? (viitab veale) Kas alamklassi indiviidid ikka kuuluvad kõik ülemklassi st kas iga alamklassi indiviid on ka tema ülemklassi indiviid?
Ontoloogia kooskõlalatus	Kas ontoloogia kooskõlalatus on kontrollitud mõne tuletusmootori abil, näiteks Pellet, Fact++ jm?
Liiasus	Kas ontoloogias esineb liiasust?
Omaduste puntrad	Kas omaduste puntrad on korrastatud? (vt punkt 6)
<i>Ontoloogia kasutatavus semantiliseks kirjelduseks</i>	
Piisavus andmeteenuste semantiliseks kirjeldamiseks	Kas ontoloogia komponendid katavad kõik andmeteenuste sisend/väljund parameetrid?
Piisavus andmekogu andmekoosseisu semantiliseks kirjeldamiseks	Kas ontoloogia komponendid katavad kõik andmekogu põhiandmete sisulised andmeväljad?

LISA 2.

Ontoloogiatehnika mõistete ja ontoloogiate loomise kohta saab lisateavet ka koolitusmaterjalidest veebilehelt <http://www.ria.ee/ontoloogia-koostamine>.

Allpool toome väikese ontoloogiate alase mõistete sõnastiku, mis võiks olla abiks käesoleva dokumendi ja teiste semantikaalaste dokumentide lugemisel.

Tabel 2. Ontoloogiatehnika põhimõistete sõnastik.

Mõiste nimetus	Mõiste nimetus inglise keeles	Selgitus
Alammõiste/alamklass	Subconcept/Subclass	Klass A on klassi B alamklass, siis kui kõik tema indiviidid kuuluvad klassi B. Alamklassid spetsialiseerivad oma ülemklasse. Näiteks, olgu klass Lapsevanem klassi Isik alamklass. See tähendab, et kõik Lapsevanem klassi indiviidid on ka Isik klassi indiviidid.
Atribuut Andmetüübimadus (OWLis)	Attribute OWL:DatatypeProperty	Atribuudid (andmetüübimadused) on spetsiifilised mingile indiviidile. Nad esitavad seoseid indiviidi ja mingi andmeväärtuse vahel. Andmetüübimadused seovad mingi indiviidi mingi andmetüübiga XML skeemi andmetüüpide hulgast (näiteks integer, float, string jms).
Funktsionaalne seos	Functional Property	Funktsionaalne seos seob ulatuse klassi indiviididest maksimaalselt 1 indiviidi (st seose kardinaalsus on 0..1).
Indiviid	Individual	Indiviidid esitavad huvivaldkonna objekte. Nad on klassi/mõiste eksemplarid ja ontoloogia baaskomponendid. Need võivad olla konkreetset objektid nagu näiteks "Hele-Mai" või siis abstraktsed objektid nagu näiteks aeg.
Kardinaalsuse kitsendus	Cardinality Restriction	Seose kardinaalsuse kitsendus määrab mitu indiviidi domeeni klassist on seotud mitme indiviidiga ulatuse klassist.
Kontseptualisatsioon	Conceptualization	Kontseptualisatsioon on inimeste maailma mõistmise ja esitamise (struktureerimise) viis. Kontseptualisatsioon on reaalsuse (või selle osa) formaalne struktuur (mingi huvivaldkonna objektid, mõisted ja teised olemid ning nende vahel kehtivad seosed), mis on mingi agendi poolt tajutud ja organiseeritud sõltumatult kasutatavast keelest ja mingi spetsiifilise situatsiooni tegelikust toimumisest. Näiteks, sõnad <i>Õun</i> ja <i>Apple</i> jagavad sama kontseptualisatsiooni.
Kooskõlalisus/ mitte vastuolulisus	Consistency	Ontoloogia on kooskõlaline siis ja ainult siis kui ta ei võimalda tuletada paikapidamatuid (kehtetuid) järeldusi
Lõikumatud klassid	Disjoint Classes	Klassid, mille indiviidide hulgad ei oma ühisosa. Mingi indiviid saab kuuluda ainult ühte nendest klassidest.
Mõiste/klass	Concept/Class OWL:Class	Mõisted (klassid) on mingis mõttes sarnaste indiviidide hulgad. Näiteks; Lapsed, Lastevanemad jms.
Ontoloogia	Ontology	Ontoloogia on reguleeritud sõnastik, mis on esitatud kirjelduskeeles, mille grammatika võimaldab sõnastiku terminite abil väljendada midagi tähenduslikku antud huvivaldkonna kohta. Grammatika sisaldab formaalseid kitsendusi ontoloogia reguleeritud sõnastiku terminite kooskasutamise kohta (ANSI/NISO Z39.19-200x). Ontoloogia defineerib (formaalselt) mingi rakendusvaldkonna kirjeldamiseks ja esitamiseks kasutatavad terminid ja nendevahelised seosed.

OWL	Ontology Web Language (OWL)	Märgendikeel ontoloogiatega realiseerimiseks. W3C standard http://www.w3.org/TR/owl-features/
Pärimine/päritavus	Inheritance	Mingi klassi alamklassi individid pärivad kõik selle klassi (ülemklassi) individide omadused.
Pöördseos	Inverse Property	Kui mingi individ a on seotud mingi seose abil individiga b , siis selle seose pöördseos seob indiviidi b individiga a . Näiteks, seos omab_lapsevanemat on seose omab_last pöördseos.
Seos Objektiomadus (OWLis)	Relation Relationship Property OWL:ObjectProperty	OWLis realiseeritud ontoloogiatega korral on seosed binaarsed ning defineeritud individide hulgal. Seosed individide vahel spetsifitseerivad kuidas 2 indiviidi on omavahel seotud. Näiteks, seos omab_last võib eksisteerida individide „Hele-Mai“ ja „Lauri“ vahel. Seosed seovad domeeni klassi kuuluvaid indiviide individidega ulatuse (muutumispirkonna) klassist. Näiteks võib individ „Hele-Mai“ kuuluda klassi Lapsevanem ja individ „Lauri“ klassi Lapsed. Seosed võivad olla kitsendatud. Neil võib olla pöördseos.
Seose domeen	Property Domain	Seose lähteklass/mõiste.
Seose kitsendused	Property Restrictions	Seosel võivad olla mitut liiki kitsendused. Näiteks, kardinaalsuse, sümmeetria, funktsionaalsuse kitsendused jm.
Seose ulatus/ muutumispirkond	Property Range	Seose sihtklass/mõiste.
Taksonoomia	Taxonomy	Mingi objektide hulga taksonoomia (taksonoomiline skeem) on klassifikatsioon, mis moodustab hierarhilise struktuuri (või üldjuhul osaliselt järjestatud hulga). Ontoloogiates võivad mõisted/klassid olla seotud taksonoomiasse.
Taksonoomiline seos	Taxonomical relationship (is-a) rdfs:subClassOf	Alamklass/ülemklass seos, mis määrab individide klassifitseerimise klassidesse.
Tuletusmootor	Reasoner	Kirjeldusloogika baasil töötav OWLi tuletusmootor (näiteks, Fact++, Pellet jms), mis võimaldab automaatselt arvutada klassifikatsiooni hierarhiat ja kontrollida ontoloogia loogilist vastuolulisust.
Täielikkus	Completeness	Ontoloogia on täielik siis ja ainult siis kui ta võimaldab tuletada kõik võimalikud kehtivad järeldused alustades ontoloogia sõnastikust ja rakendades lubatud tuletusreegleid
Ülemmõiste/ülemklass	Superconcept Superclass	Ülemklass on oma alamklasside üldistus.