

LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

**CEĻĀ UZ SEMANTISKĀM ZINĀŠANĀM:
VESELĪBAS APRŪPES ONTOLOĢIJU SEMANTISKAIS
NOVĒRTĒJUMS**

MAGISTRA DARBS

Autors: Ernests Madelāns

Stud. apl. Nr. em19065

Darba vadītājs: Doc., Dr. Sc. Comp. Edgars Rencis

RĪGA 2023

ANOTĀCIJA

Pēdējo trīsdesmit gadu laikā pasaulē ir parādījušies arvien vairāk neviendabīgi veselības aprūpes dati. Tāpēc biomedicīnas ontoloģijas kopā ar to terminoloģijām spēlē arvien lielāku lomu veselības aprūpes zināšanu apstrādē, pārvaldībā un atspoguļošanā. Taču ontoloģiju izmantošanas panākumi ir lielā mērā atkarīgi no to kvalitātes, un bieži ontoloģiju kvalitāte nav pieņemama.

Ontoloģijas kvalitātes novērtēšana ir neatņemama ontoloģijas veidošanas un uzturēšanas sastāvdaļa. Tā tiek veikta gan ontoloģiju izstrādes laikā, gan pirms to publicēšanas, gan arī pirms to lietošanas, lai identificētu neatbilstības un modelēšanas kļūdas.

Šajā darbā tiek atkārtota pieeja vienam no pēdējo gadu ievērojamākajiem pētījumiem ontoloģiju novērtēšanas jomā, taču daudz reālistiskākā vidē. Salīdzinājumā ar minēto pētījumu, tiek izmantoti brīvi pieejami dati no Nacionālā Biomedicīnas ontoloģijas centra (NCBO) bioportāla - lielākās biomedicīnas ontoloģiju krātuves. Tiek izstrādātas 20 dažādas prasības, lai validētu un verificētu 10 dažādas ontoloģijas ar četriem ontoloģiju novērtēšanas rīkiem (Protege, TDDOnto, OOPs! un Themis). Visbeidzot, tiek salīdzināti iegūtie rezultāti un secināti šīs pieejas iespējamie izmantošanas scenāriji un praktiskums.

Atslēgvārdi: veselības aprūpe, ontoloģijas novērtējums, zināšanu modelēšana, saistītie dati, testa gadījumi

ABSTRACT

Thesis title: Towards Semantic Knowledge: Semantic Evaluation of Healthcare Ontologies

Over the past thirty years, large amount of heterogeneous health care data has emerged around the world. That is why biomedical ontologies together with their terminologies play an increasing role in the processing, management, and representation of healthcare knowledge. However, the success of using ontologies largely depends on their quality, and often the quality of ontologies is not acceptable.

Ontology quality assessment is an integral part of ontology development and maintenance. It is conducted during the development of ontologies, before their publication, and before their use, to identify possible inconsistencies and modeling errors.

This work repeats the approach of one of the most significant studies in the field of ontology evaluation in recent years, but in a much more realistic setting. Compared to the mentioned study, data is used from the National Center for Biomedical Ontology (NCBO) BioPortal, the largest repository of freely available biomedical ontologies. 20 different requirements are developed to validate and verify 10 different ontologies with four ontology evaluation tools (Protege, TDDOnto, OOPs! and Themis). Finally, the obtained results are compared, and the possible use scenarios and practicality of this approach are concluded.

Keywords: healthcare, ontology evaluation, knowledge modeling, linked data, test cases

AUTOREFERĀTS

Šajā darbā autors ir aprakstījis apjomīgu literatūras apskatu par visām ar ontoloģiju novērtēšanu saistītām tēmām, kuras ir nepieciešams zināt pirms galveno darba mērķu sasniegšanas. Darbā ir veikta ļoti detalizēta dažādu ontoloģiju novērtēšana ar vairākiem ontoloģiju novērtēšanas rīkiem. Visu testu rezultāti ir nodokumentēti un smalki aprakstīti.

Darba novitāte. Vēl nesen neeksistēja pētījums ontoloģiju novērtēšanas jomā, kas mēģinātu apvienot vairākus ontoloģiju novērtēšanas rīkus, lai validētu un verificētu ontoloģiju kvalitāti. 2020. gadā šāds pētījums tika veikts, taču diezgan ierobežotā vidē [60]. Šajā darbā autors ir atkārtojis minēto pieeju daudz reālistiskākā vidē, izmantojot ontoloģijas no Nacionālā Biomedicīnas ontoloģijas centra (NCBO) bioportāla, kurā ir brīvi pieejamas simptiem dažādas biomedicīnas ontoloģijas.

Literatūras izpēte. Darba autors ir apskatījis ļoti lielu apjomu ar dažādiem zinātniskiem rakstiem. Literatūras apskatā tiek detalizēti aprakstīti semantiskie modeļi, ontoloģiju novērtēšanas standarti un metodes, kā arī pēdējo divdesmit gadu laikā izveidotie ontoloģiju novērtēšanas rīki. Turklāt tiek apskatīti pētījumi par NCBO bioportāla ontoloģiju kvalitāti un detalizēti apskatīta ontoloģiju terminoloģija. Lielākā daļa no izmantotās literatūras avotiem nāk no zinātniskām publikācijām.

Izklāsta pamatīgums. Šajā darbā autors ir virzījies no vienkāršas ontoloģijas definīcijas līdz pamatīgam ontoloģiju terminoloģiju izklāstam. Autors ir dziļi iedziļinājies katrā no galvenajām tēmām, kas nepieciešamas, lai pilnīgi izprastu visus ontoloģiju novērtēšanas elementus un spētu paveikt nepieciešamos uzdevumus.

Paveiktā praktiskā darba apjoms. Darbā ir ieguldītas būtiskas pūles ontoloģiju novērtēšanas procesā. Ir apskatītas 10 dažādas ontoloģijas ar kurām autors ir iepazinies un kuras ir novērtētas ar četriem dažādiem rīkiem. Katrai no 10 ontoloģijām ar 3 novērtēšanas rīkiem autors ir manuāli pārbaudījis 20 dažādas prasības.

Rezultātu aprobācija. Šajā darbā izmantotā metodoloģija (t. sk. izmantotie ontoloģiju novērtēšanas rīki un ontoloģiju novērtēšanas prasības) ir balstīta uz citiem zinātniskiem rakstiem, tāpēc rezultātu pieraksts līdzinās citiem zinātniskiem rakstiem šajā jomā. Rezultātus ir iespējams samērot ar citiem pētījumiem pie nosacījuma, ka tajos ir izmantotas tās pašas ontoloģijas.

Darba noformējuma kvalitāte. Darba autors ir vairākkārt pārlasījis šo darbu un novērsis visas atrastās gramatikas kļūdas. Darba noformēšana ir veikta izmantojot visjaunāko pieejamo maģistra darba izstrādes un aizstāvēšanas metodisko norādījumu dokumentu. Darbā ir izmantota oficiāli pieņemtā nozares terminoloģija. Visiem terminiem, kuriem nav bijis pieejams latviskais tulkojums, autors ir pats izvēlējis latvisko tulkojumu, blakus (iekavās) pievienojot termina nosaukumu angļu valodā.

Plāģiāta iespējamība. Autors visām aizgūtām idejām, definīcijām, formulējumiem un attēliem ir pievienojis atsauces. Visi darbā izmantotie burtiskie tulkojumi un tuvi pārstāsti ir atzīmēti kā teksta aizguvumi.

SATURS

APZĪMĒJUMU SARAKSTS.....	8
IEVADS	9
1. ONTOLOĢIJAS UN SEMANTISKAIS TĪMEKLIS	11
1.1. Kas ir lietu internets?	11
1.2. Kas ir ontoloģijas?	12
1.3. Ontoloģiju loma veselības aprūpē.....	13
1.4. Vai dažādu domēnu ontoloģijas atšķiras?.....	14
1.5. Kas ir ontoloģiju novērtēšana?	14
1.6. Kas ir semantiskais tīmeklis?.....	14
2. MOTIVĀCIJA.....	16
3. LITERATŪRAS APSKATS	19
3.1. Semantiskie modeļi.....	19
3.2. Ontoloģiju kvalitātes novērtēšana.....	20
3.3. Ontoloģiju novērtēšanas veidi	21
3.4. Ontoloģijas un to novērtēšanas rīki	25
3.5. NCBO bioportāls	29
3.6. NCBO bioportāla ontoloģiju novērtēšanas kvalitātes pārskats.....	30
3.7. Ontoloģiju kvalitātes nodrošināšanas pārskats	31
4. TERMINOLOĢIJA	33
4.1. Ontoloģijas.....	33
4.2. OWL	33
4.3. Fakti	34
4.4. Aksiomas	34
4.4.1. Klases aksiomas	35
4.4.2. Īpašības aksiomas	36
4.5. Anotācijas	36
4.6. Entītijas	36
4.6.1. Individu entītijas	36
4.6.2. Klašu entītijas	37
4.6.3. Īpašību entītijas.....	37
4.6.4. Ontoloģiju entītijas	38

4.7. Saistība.....	38
4.8. Argumentēšana	38
5. TEORĒTISKAIS IETVARŠ	40
5.1. Konceptuālais modelis	40
6. METODOLOĢIJA	42
6.1. Izvēlētās ontoloģijas	42
6.2. Ontoloģiju novērtēšana	44
6.3. Darbā izmantotie ontoloģiju novērtēšanas rīki	45
6.3.1. <i>Themis</i>	45
6.3.2. <i>TDDOnto</i>	47
6.3.3. <i>OOPs!</i>	49
6.3.4. <i>Protege</i>	51
7. ONTOLOĢIJU NOVĒRTĒŠANA.....	53
7.1. Novērtēšana ar <i>Themis</i>	53
7.2. Novērtēšana ar <i>OOPs!</i>	56
7.3. Novērtēšana ar <i>TDDOnto</i>	58
7.4. Novērtēšana ar <i>Protege</i>	58
REZULTĀTI	62
SECINĀJUMI	64
IZMANTOTĀ LITERATŪRA UN AVOTI	67
PIELIKUMI.....	76

APZĪMĒJUMU SARAKSTS

ABox Apgalvojumu kaste - (Assertion box)

DAML+OIL Semantiskā iezīmēšanas valoda tīmekļa resursiem - (Semantic markup language for Web resources)

FAIR data Atrodami, pieejami, sadarbspējīgi un atkārtoti lietojami dati - (Findable, Accessible, Interoperable, and Reusable data)

IoT Lietu internets - (Internet of Things)

IRI Internacionalizēts resursu identifikators - (Internationalized Resource Identifier)

JAR Java arhīva fails - (Java archive file)

OWL Tīmekļa ontoloģijas valoda - (Web Ontology Language)

RDF Resursu apraksta ietvars - (Resource Description Framework)

RDFs Resursu apraksta ietvara shēma - (Resource Description Framework Schema)

SPARQL RDF vaicājumu valoda - (RDF Query Language)

SWRL Semantiskā tīmekļa kārtulu valoda - (Semantic Web Rule Language)

TBox Terminoloģiskā kaste - (Terminological box)

URI Vienotais resursu identifikators - (Uniform Resource Identifier)

IEVADS

Pēdējo divdesmit līdz trīsdesmit gadu laikā pasaulē ir parādījušies arvien vairāk nevienmērīgi veselības aprūpes dati. Tieši tādēļ biomedicīnas ontoloģijas kopā ar to terminoloģijām spēlē arvien lielāku lomu veselības aprūpes zināšanu apstrādē, pārvaldībā un atspoguļošanā. Parādās arvien vairāk dažādas sistēmas, kurās šīs ontoloģijas tiek izmantotas visdažādākajiem mērķiem. Taču ontoloģiju izmantošanas panākumi ir lielā mērā atkarīgi no to kvalitātes. Datu kvalitātes jautājumi nevar tikt atrisināti ar ontoloģiju palīdzību gadījumos, kad pašu ontoloģiju kvalitāte nav pieņemama.

Šajā darbā tiek apskatītas dažādas ontoloģiju novērtēšanas pieejas. Detalizētāk tiek apspriesta viena konkrēta ontoloģiju novērtēšanas pieeja, kas ir pēc šīs pieejas autoru [60] domām milzīgs solis uz priekšu ontoloģiju novērtēšanas jomā, jo nekas līdzīgs līdz šim netika darīts. Tālāk tiek izvēlētas vairākas NCBO bioportālā brīvi pieejamas ontoloģijas, kā arī tiek izvēlēti dažādi ontoloģiju novērtēšanas rīki. Tiek mēģināts pielietot šo ontoloģiju novērtēšanas pieeju, kā arī tiek mēģināts novērtēt šīs pieejas iespējamību un praktiskumu praksē.

Šī darba galvenie mērķi un to ieguldījums:

1. mērķis: identificēt un apskatīt esošās ontoloģiju novērtēšanas pieejas un tajās izmantotos ontoloģiju novērtēšanas rīkus.
1. ieguldījums: Sistemātiski aplūkota eksistējošā literatūra par ontoloģiju novērtēšanas tehnikām un ontoloģiju novērtēšanas rīkiem.
2. mērķis: izpētīt sistēmas un rīkus, kas ļauj novērtēt veselības aprūpes ontoloģiju kvalitāti un pamēģināt tos pielietot praksē.
2. ieguldījums: šajā darbā autors ir apskatījis un pielietojis praksē vienu ontoloģiju novērtēšanas metodoloģiju, kas līdz šim ir tikai vienreiz parādījusies eksistējošā literatūrā.

Šī darba sadaļas ir sakārtotas sekojoši:

1. nodaļā ir aprakstīts ievads par ontoloģijām, semantisko tīklu un to izmantošanu mūsdienās;
2. nodaļā ir detalizēti aprakstīta šī darba motivācija;
3. nodaļā ir sniegts literatūras apskats par dažādām ontoloģiju novērtēšanas pieejām (ieskaitot dažādus ontoloģiju novērtēšanas rīkus un arī pašas veselības aprūpes ontoloģijas);
4. nodaļā ir skaidrota ar ontoloģijām saistītā terminoloģija;
5. nodaļā ir aprakstīts teorētiskais ietvars šī darba praktiskās daļas tapšanai;
6. nodaļā ir detalizēti aprakstīta metodoloģija (ieskaitot izvēlētās ontoloģijas, izvēlētos ontoloģiju novērtēšanas rīkus un darba gaitu);
7. nodaļā ir aprakstīta ontoloģiju novērtēšana ar katru no darbā izmantotajiem ontoloģiju novērtēšanas rīkiem;
8. nodaļā ir aprakstīti šī darba rezultāti;
9. nodaļā ir aprakstīti šī darba secinājumi.

1. ONTOLOĢIJAS UN SEMANTISKAIS TĪMEKLIS

Veselības aprūpe globāli ir milzīga nozare un Amerikā vien 2021. gadā veselības aprūpes izdevumi pārsniedza 4 triljonus ASV dolāru. Tās izdevumi veido gandrīz piektdaļu no visa ASV iekšzemes kopprodukta [16]. Pēdējo gadu laikā daudzi globālie trendi (covid-19, globālā klimata sasilšana) ir izaicinājuši gan šo nozari, gan arī pašu cilvēku veselību [3, 57].

Pasaules iedzīvotāju vidējais vecums strauji aug un tiek lēsts, ka līdz 2050. gadam vairāk kā 2 miljardi cilvēku būs sasnieguši 60 gadu vecumu vai vairāk. Tāpat tiek lēsts, ka līdz 2030. gadam mirstība no hroniskām slimībām palielināsies līdz 66%. Bezvadu tehnoloģiju sasniegumi ir spējīgi pozitīvi ietekmēt hronisku slimību gaitu, pateicoties to spējām ātri atklāt šīs slimības [57, 65].

Bezvadu tehnoloģijām un īpaši lietu internetam (angļu: Internet of Things, IoT) ar semantiskajiem modeļiem ir lielas iespējas uzlabot mūsdienu veselības aprūpi. Šīm tehnoloģijām ir spējas pozitīvi ietekmēt pacientu aprūpi, racionalizēt procesus, kā arī samazināt izmaksas. Veselības aprūpes speciālistiem ir nepieciešami reāllaika dati, lai spētu pieņemt labākus lēmumus. Tāpat arī pacientu uzraudzība un līdzekļu izsekošana var tikt uzlabota ar šīm tehnoloģijām [1, 44].

1.1. Kas ir lietu internets?

Lietu internets jeb IoT raksturo konceptu, kura ietvaros visas ierīces ir savā starpā savienotas, un vienlaicīgi tās ir savienotas arī ar internetu. IoT galvenais mērķis ir padarīt ierīces savstarpēji saistītas un vienlaicīgi vākt un apkopot visus ierīču datus. Pēdējos gados ir strauji pieaugusi pētniecības kopienas interese pret IoT tehnoloģijām, kas savukārt paātrināja dažādu, ar IoT saistītu, lietojumprogrammu izveidi [61].

Mūsdienās vākt datus un apmainīties ar tiem spēj ne tikai mobilās ierīces vai datortehnika, bet arī ne tik acīmredzama sadzīves tehnika kā ledusskapji vai putekļu sūcēji. Tāpēc nav pārsteigums, ka par IoT tehnoloģijām sāk interesēties arvien vairāk speciālisti no ļoti dažādām nozarēm kā, piemēram, laikapstākļu monitorings, energoapgāde, izglītība, satiksmes kontrole un citām [61]. Nav noslēpums, ka arī veselības nozarē arvien pieaug ontoloģiju virzītu e-veselības risinājumu attīstība, kuru funkcionalitāte tiek uzlabota ar IoT tehnoloģijām [7, 42, 57]. Piemēram, pašreizējās elektrokardiogrāfijas pieejās tiek izmantotas tādas ierīces, kuras pacientam tiek uzvilktas virsū, un tās reāllaikā uzrauga pacienta sirds veselību. Šāda pieeja padara vienkāršāku pacienta uzraudzību un automātiski identificē neparastu sirds uzvedību [42].

IoT tehnoloģijas ļauj datu apmaiņai notikt starp dažādām sensoru ierīcēm reāllaikā. Tie var būt gan tādi dati kā, piemēram, veselības vēsture, gan arī dažāda lietotāju informācija, vai kāda cita veselības jomai specifiska informācija. Tiwari un Abraham [60] raksta, ka lai informācija varētu tikt uzģenerēta viedās veselības aprūpes sistēmā un mijiedarboties ar dažādām ierīcēm nederēs jebkāda datu struktūra. Šādos gadījumos ir nepieciešama specifiska datu struktūra.

1.2. Kas ir ontoloģijas?

Mūsdienās vārds “ontoloģija” tiek izmantots vairākos veidos un tam var atrast vairākas definīcijas. Viena no definīcijām, kas mūsdienās tiek bieži lietota ir atrodama te [25]. Tā sekojoši skaidro zināšanu inženierijas kopienā lietoto vārda “ontoloģija” būtību:

“Ontoloģija ir skaidra konceptualizācijas specifikācija. Termins ir aizgūts no filozofijas, kur ontoloģija ir sistemātisks esamības izklāsts. AI sistēmām “pastāv” tas, ko var attēlot.” [25]

Tāpat, lai izskaidrotu, kas ir ontoloģija, daudzos darbos tiek minēts arī darbs [28] “Kas ir ontoloģija?”, kurā autori spriež par mūsdienās eksistējošām divām populārām ontoloģijas nozīmēm. Kur pirmā ir filozofiskā nozīme, kas sev līdzī nes ļoti vecas tradīcijas. Taču otrā ir nesen zināšanu inženierijas kopienā parādījusies nozīme, kura apzīmē “precīzas konceptualizācijas specifikācijas”. Šis savukārt ir saīsinājums garajai definīcijai, ko viens no darba [28] autoriem izveidoja jau 1998. gadā (balstoties uz [10]):

“Ontoloģija ir loģiska teorija, kas atspoguļo formālās vārdnīcas paredzēto nozīmi, t.i., tās ontoloģisko saistību ar noteiktu pasaules konceptualizāciju. Paredzētos loģiskās valodas modeļus, kas izmanto šādu vārdu krājumu, ierobežo tās ontoloģiskā apņemšanās. Ontoloģija netieši atspoguļo šo apņemšanos (un pamatā esošo konceptualizāciju), tuvinot šos paredzētos modeļus.” [10]

Lai izprastu galvenos ontoloģiju pielietošanas veidus var ieskatīties Gruber [26] darbā, kur tie ļoti īsi un precīzi tiek aprakstīti. Balstoties uz Gruber [26], ontoloģiju galvenā loma ir padarīt datus eksportējamus, tulkojamus un apvienojamus starp dažādām neatkarīgi izstrādātām sistēmām. Turklāt, Gruber [26] jau 2009. gadā norādīja, ka veiksmīgs ontoloģiju pielietošanas rezultāts ir,

piemēram, savietojamība starp dažādām datu bāzēm, meklēšana starp dažādām datu bāzēm, vai tīmekļa pakalpojumu integrācija.

Taču mūsdienās pieaug ne tikai ontoloģiju definīciju daudzums, bet arī pašu ontoloģiju daudzums, kā arī to apjomi un arī sarežģītības pakāpe. Ir skaidrs, ka ontoloģiju veidošana bez iespējas tām savā starpā dalīties ar kopīgām zināšanām būtu bezjēdzīgs process. Mūsdienās ontoloģiju izstrādātāji cenšas panākt to pēc iespējas lielāku savietojamību, kas līdz šim joprojām ir eksistējoša ontoloģiju problēma [10, 60]. Rezultātā, jo sarežģītākas kļūst ontoloģijas, jo biežāk to kvalitāte samazinās. Un, jo sliktāka ir ontoloģiju kvalitāte, jo mazāka iespēja ir to pilnvērtīgi izmantot tālāk. Protams, mūsdienās eksistē vairākas ontoloģiju veidošanas metodoloģijas, taču tās nepalīdz vairumā gadījumu, kad tiek lemts, kuru ontoloģiju labāk izvēlēties atkārtotai izmantošanai [10].

1.3. Ontoloģiju loma veselības aprūpē

Biomedicīnas ontoloģijas spēlē svarīgu lomu veselības aprūpes sistēmās ar to spējām kodēt diagnozes, laboratorijas testus, veselības kartes, kā arī citu veselības aprūpes informāciju [7].

Mūsdienās milzīgu datu kopu integrācija dažādiem jomas speciālistiem sniedz milzīgas iespējas jaunu zināšanu atklāšanā, un ontoloģijas spēlē svarīgu lomu šo zināšanu organizēšanā [7, 60]. Dati tiek organizēti visdažādākajos veidos. Tie var būt organizēti, piemēram, kā aksiomas vai kā instances, vai arī kā koncepti un attiecības. Veselības aprūpes ontoloģijām ir liela loma zināšanu attēlošanā un datu integrēšanā. Veselības aprūpē tiek izmantotas dažādas sistēmas, starp kurām datu apmaiņa notiek pateicoties IoT. Savukārt ontoloģijas sniedz iespēju reprezentēt visu šo milzīgo informācijas apjomu un attēlot visdažādākās veselības aprūpes zināšanas [60].

Izmantojot ontoloģijas, veselības aprūpes jomā ir iespējams organizēt veselības aprūpes informāciju un tālāk šo informāciju pārveidot. Ontoloģijas ļauj strādāt ar tādu informāciju, kas, piemēram, ir veselības karšu ieraksti, laboratorijas testi, pacientu diagnostika, kā arī veselības resursi. Ontoloģijās konceptu attiecības ir hierarhiskā formā, un šāds formāts palīdz ne tikai integrēt datus, bet arī tam ir svarīga loma zināšanu reprezentēšanā [60].

1.4. Vai dažādu domēnu ontoloģijas atšķiras?

Nemot vērā to, ka šajā darbā autors ir koncentrējies tieši uz biomedicīnas jomu, šī jomas ontoloģijām un to problēmām, tad, pēc autora domām, ir svarīgi saprast to, ka visu jomu ontoloģijas nav vienādas. Jautājums saistībā ar dažādu domēnu ontoloģiju atšķirībām ir izcili atbildēts šeit [44]. Balstoties uz Nikiforova et al. [44], galvenā atšķirība starp biomedicīnas ontoloģijām un tipiskām ontoloģijām, kas veidotas plašai izmantošanai, ir tāda, ka biomedicīnas ontoloģijās par prioritāti ir izvirzīta datu koplietošana un savietojamība. Biomedicīnas ontoloģijas (arī citu domēnu ontoloģijas) bieži tiek izstrādātas balstoties uz tūlītējiem analīzes mērķiem un tā ir šo ontoloģiju unikālā iezīme. Taču tas nozīmē arī to, ka bieži šāda veida ontoloģijas laika gaitā tiek paplašinātas un attīstītas tālāk, iekļaujot datus no citām ontoloģijām. Šādā veidā ontoloģiju zināšanas tiek padarītas izteismīgākas un pilnīgākas, taču daži autori [29, 47] secina, ka tas negatīvi ietekmē šo ontoloģiju kvalitāti.

1.5. Kas ir ontoloģiju novērtēšana?

Tradicionāli ontoloģiju novērtēšana apzīmē procesu, kura ietvaros tiek noteikta ontoloģijas kvalitātes atbilstība vērtēšanas kritērijiem. Atkarībā no dažādām ontoloģijām un to mērķiem kritēriji var būt dažādi [7]. Dažādu zinātnisko publikāciju autoru virzieni mēdz atšķirties, taču lielākoties šīs pieejas iekrīt kādā no 4 dažādām kategorijām. Daži autori ontoloģijas salīdzināta ar “zelta standartu”, kas var būt gan cita ontoloģija, gan arī jomas ekspertu sagatavota, vai no vairākiem dokumentiem apkopota konceptu reprezentācija. Citi pēta rezultātus pēc ontoloģijas lietošanas lietojumprogrammā. Citos pētījumos tiek salīdzināts ar datu avotu par konkrēto jomu, kura tiek ontoloģijā izmantota. Taču dažādos citos pētījumos cilvēki vērtē to, cik ļoti ontoloģija atbilst prasībām un standartiem [12]. Ontoloģiju kvalitātes novērtēšana ir ļoti svarīgs process. To darot var pamanīt gan loģiskās, gan arī sintaktiskās kļūdas, kuras mēdz dēvēt par ontoloģiju neparedzētām sekām [60].

1.6. Kas ir semantiskais tīmeklis?

Viens no mūsdienu lielākajiem izaicinājumiem IoT tehnoloģiju un pieejas adaptācijai ir šo pašu tehnoloģiju straujā izaugsme. Līdz ar dažādu IoT tehnoloģiju parādīšanos, strauji pieaug arī šo tehnoloģiju datu daudzveidība. Dati tiek veidoti un glabāti ne tikai ar dažādu saturu, bet arī ar

atšķirīgu struktūru. Tie var būt gan strukturēti, gan nestrukturēti, gan arī daļēji strukturēti dati. Turklāt, glabātajos datos atšķiras gan datu valodas, gan mērvienības, gan arī paši datu avoti. Kamēr liela daļa datu mūsdienu tīmeklī tiek atspoguļoti cilvēkam saprotamā veidā, ierīces šos datus nesaprot un lasīt neprot [44]. Mishra un Jain [61] raksta, ka šo problēmu spēs atrisināt semantiskais tīmeklis (angļu: semantic web, Web 3.0). Semantiskā tīmekļa galvenais uzdevums ir atvieglot dažādu ierīču dzīvi, padarot tīmekļa datus mašīnlasāmus un veicinot sadarbību starp dažādām IoT ierīcēm. Savukārt semantiskās pieejas rūpējas par to, lai padarītu datu meklēšanu vienkāršāku un atvieglotu apjomīgu datu kopu lasīšanu. Vairāki autori [44, 61] uzskata, ka tas viss ir iespējams jau tagad, pateicoties ontoloģijām un to dažādiem formātiem, jo tās izcili spēj standartizēt terminoloģiju un noteikt attiecības starp dažādiem elementiem. Ar ontoloģiju palīdzību jebkurus datus var padarīt mašīnlasāmus, jo tās izmanto mašīnsaprotamu (angļu: machine-understandable) pieeju, aprakstos katru entītijū, attiecības un indivīdus. Līdz ar to, ontoloģijas spēlē vienu no galvenajām lomām, tverot un apkopojot datu semantiku nepārtraukti augošajā semantiskajā tīmeklī.

2. MOTIVĀCIJA

Mūsdienu veselības aprūpes jomā līdz šim ir investēts ļoti daudz līdzekļu, lai izveidotu visaptverošus un atkārtoti lietojamus informācijas resursus, kas spētu efektīvi palīdzēt ar datu izmantošanu un analīzi [63]. Taču pēdējos gados veselības aprūpes datu savietojamība kļūst par arvien lielāku problēmu. Dati nāk no vairākiem avotiem un dažādos formātos, tāpēc dažādas institūcijas saskaras ar tādu problēmu, ka datus ir grūti integrēt to iekšējās sistēmās [44, 51, 63].

Balstoties uz vairākiem pētījumiem [44, 51, 63], ir skaidrs, ka veselības aprūpē joprojām trūkst tādi datu resursi, kas būtu pilnībā atrodami, pieejami, sadarbspējīgi un atkārtoti lietojami, un tāpēc atbilstu FAIR (angļu: Findable, Accessible, Interoperable, Reusable, FAIR) principiem.

Spilgtus piemērus datu savietojamības problēmai var atrast arī te pat Latvijā, piemēram, Latvijas Biomedicīnas pētījumu un studiju centrā (LBMC). Balstoties uz Nikiforova et al. [44], LBMC lieto datus no dažādiem avotiem, piemēram:

- Latvijas Nacionālās Biobankas, kas ietver sevī Valsts iedzīvotāju Genoma datubāzi, kas uzglabā un apstrādā Latvijas iedzīvotāju veselības informācijas datus un bioloģisko paraugu kolekciju;
- Slimību profilakses un kontroles centra Veselības statistikas datubāzi, kur ir apskatāmi dati par Veselības aprūpi un to resursiem, Saslimstību un slimību izplatību, Mirstību, Medicīnas tūrismu un citi dati;
- Doktorantūras studentu iegūtie dati no visdažādākajām anketām par dažādām tēmām;
- Dati pieejami dažādās citās uzņēmumu vai valsts datubāzēs;
- Dati no starptautiskām sistēmām, kas ietver sevī, piemēram, dažādas medicīnas iestādes.

Šie visi dati nāk visdažādākās formās. Tie varbūt dati no roku rakstītām anketām, dati no digitālām aptaujām, dati no digitālām vai ar roku pildītām veidlapām, dati no dažādām medicīnas ierīcēm (piemēram, ultrasonogrāfijas vai elektrokardiogrāfijas dati), utt. Nav grūti secināt, ka rezultātā rodas dažādas, ar datiem saistītas, problēmas. Problēmas rodas ar datu tipiem un formātiem, tiek izmantotas nepiemērotas sistēmas un nepareizi datu bāzu dizaini. LBMC nav vienota risinājuma šai problēmai, un tā tiek risināta atkarībā no situācijas un projekta vajadzībām, izmantojot dažādas esošās pieejas [44].

Nikiforova et al. [44] secina, ka vispiemērotākā pieeja šai datu savietojamības problēmai esot ontoloģijas un semantiskie modeļi. Īpaši piemēroti tie ir milzīgu datu kopu un lielu organizāciju gadījumos. Tas ir tādēļ, ka ontoloģijas strādā kā terminoloģiju standartizētājs, nosakot attiecības

starp aplūkotajiem elementiem. Šādā veidā ne tikai tiek atvieglots darbs ar datiem pašiem pētniekiem un zinātniekiem, bet arī tiek vienkāršota pieeja neatrisinātu problēmu apkarošanai.

Mūsdienās eksistē daudz ontoloģijas, kuras tiek semantiski modelētas. Arī veselības aprūpē var atrast simtiem ontoloģiju, kas palīdz integrēt veselības aprūpes informāciju un ir savienojamas ar dažādiem veselības aprūpes rīkiem [60]. Taču Latvijā ontoloģiju izmantošana medicīnā vēl nav tik populāra, tāpēc ir svarīgi saprast ontoloģiju izmantošanas iespējas un šādas izmantošanas lielākos stūrakmeņus.

Kā jau tika minēts, ontoloģijas novērtēšana ir ontoloģijas izstrādes un uzturēšanas neatņemama sastāvdaļa. Balstoties uz Tiwari un Abraham [60], lielākā daļa no veselības aprūpes ontoloģijām tiek novērtētas ar kādu no trīs veidiem. Tas ir vainu Semantiskās tīmekļa kārtulas valodas (SWRL) noteikumi, SPARQL valoda, vai arī iekšā iebūvēts argumentētājs. Tas nozīmē, ka šīs ontoloģijas tiek pārbaudītas tikai pret atbilstību, konsekveni un klasifikāciju. Līdz ar to mūsdienu literatūrā pietrūkst veselības aprūpes ontoloģiju novērtēšanas veids, kas spētu veselības aprūpes ontoloģijai novērtēt tās zināšanu pilnību.

Tiwari un Abraham [60] ir pirmie, kas izmanto vairākus dažādus ontoloģiju novērtēšanas rīkus, lai analizētu ontoloģijas kvalitāti. Autori savā darbā izmanto 4 dažādus ontoloģiju novērtēšanas rīkus. Tie ir OOPS! un Themis, kas abi ir tīmekļa rīki, Protégé, kas ir ontoloģijas redaktors ar argumentētāju, kā arī TDDonto, kas ir Protégé spraudnis. Taču minētajiem autoriem ir diezgan ierobežots skatījums uz šo pieeju. Pirmkārt, autoru darbā tiek novērtēta tikko izveidota SHCO ontoloģija, kas veidota pēc jaunākajiem ontoloģiju veidošanas standartiem. Otrkārt, balstoties uz autoru sniegto SHCO grafisko attēlojumu, ir skaidrs, ka SHCO ontoloģijai ir ļoti vienkārša struktūra. Treškārt, autori salīdzina SHCO ar 8 citām ontoloģijām no kurām 7 nav pieejamas tiešsaistē, un 5 no šīm 8 ontoloģijām ir veidotas atkārtoti neizmantojot nevienu citu ontoloģiju. Rezumējot, šādas pieejas lielākā problēma ir tāda, ka tā tiek veikta ļoti ierobežotā un sterilā vidē.

Šajā darbā tiek atkārtota Tiwari un Abraham [60] darba pieeja, taču daudz reālistiskākā vidē. Ontoloģiju meklēšanai tiek izmantots NCBO bioportāls – viena no pasaulē lielākajām biomedicīnas ontoloģiju krātuvēm. Tiek izmantotas brīvi pieejamas ontoloģijas ar salīdzinoši lieliem šo ontoloģiju apmeklējumu skaitiem. Turklāt, visas no šīm ontoloģijām ir veidotas, otrreizēji importējot kādu citu jau eksistējošu ontoloģiju, kas padara šīs ontoloģijas daudz apjomīgākas un vērtēšanā sarežģītākas. Visbeidzot, paturot prātā to, ka Latvijas veselības aprūpē

ontoloģijas un to novērtēšana netiek plaši izmantota praksē, ir vērtīgi pārbaudīt šīs pieejas praktisku pielietojumu, kā arī secināt tās pieejamību, lietošanas vienkāršību un lielākos stūrakmeņus.

3. LITERATŪRAS APSKATS

Šī darba literatūras apskats sastāv no vairākām daļām. Pirmkārt, tiek prezentēts īss apraksts par semantiskajiem modeļiem. Otrkārt, tiek apskatīti dažādi veidi tam kā mūsdienās tiek novērtēta ontoloģiju kvalitāte. Treškārt, tiek apskatīti dažādi iepriekšējie akadēmiskie pētījumi par ontoloģiju novērtēšanas metodēm. Ceturtkārt, autors apskata mūsdienās izmantojamo ontoloģiju novērtēšanas rīkus. Tālāk tiek aprakstīta viena no lielākajām biomedicīnas ontoloģiju glabātuvēm. Visbeidzot, tiek īsi apskatītas ontoloģiju kvalitātes nodrošināšanas pieejas.

3.1. Semantiskie modeļi

Mūsdienās veselības aprūpes dati tiek izmantoti dažādiem mērķiem. Piemēram, tie tiek izmantoti lēmumu pieņemšanā, darbplūsmu veidošanā un citām dažādām aktivitātēm. Lai visas šīs aktivitātes būtu efektīvas ir svarīgas gan platformas, gan arī datu apmaiņas un ietvaru standartizācija. Lai nodrošinātu kvalitatīvu viedo veselības aprūpes platformu darbošanos ir jāstājas priekšā izaicinājumiem, kas nāk no lietotāja interfeisa puses, autentifikācijas un pacienta datu drošības [1]. Ir pieejami dažādi standarti un pieejas kā cīnīties ar šīm problēmām, taču šie standarti ir atkarīgi no organizācijām, kas nosaka šo ziņošanas formātu. Semantisko modeļu priekšrocība ir tāda, ka tie spēj nodefinēt dažādu veidu ziņojumu formātu. Līdz ar to jebkāda standarta vai ietvara vietā ir iespējams izmantot semantisko modeli, lai sistēmā nodefinētu specifisku protokolu starp viedās veselības ierīci un serveri [33].

Semantiskiem modeļiem ir iespējami dažādi izmantošanas veidi. Ar tiem var grupēt un filtrēt dažādus datus, var definēt noteikumus vai prezentēt informāciju [33]. Piemēram, Dawood un Sah [18] savā darbā prezentē veselības aprūpes informācijas sistēmas modeli, kas ar semantisko tīmekļa tehnoloģiju palīdzību apraksta ierīces datus un definē noteikumus, lai spētu izdarīt korektus secinājumus par veselības aprūpes pakalpojumiem. Šādā modelī veselības aprūpes dati tiek ievākti no dažādām ierīcēm un var tikt otrreizēji izmantoti ne tikai tajā ierīcē vai organizācijā, kas šos datus ir izveidojuši. Ar semantisko tīmekļa tehnoloģiju palīdzību ir iespējams piekļūt šiem datiem citādos veidos, piemēram, no citiem autorizētiem servisiem. Līdz ar to dažādu sistēmu dati var tikt izmantoti otrreizēji ar citām sistēmām.

Sotiriadis et al. [55] ir izveidojuši viedo pacientu uzraudzības sistēmas arhitektūru. Šāda sistēma izmanto semantisko tīmekli un elektroniskos veselības ierakstus priekš bipolāriem pacientiem. Autori apgalvo, ka semantiskais tīmeklis ļauj sistemātiski iegūt zināšanas kā arī veidot

uz šīm zināšanām balstītas ontoloģijas. Turklāt ar semantiskā tīmekļa palīdzību ir iespējams analizēt pacientu ierakstus un filtrēt uz pierādījumiem balstītas vadlīnijas. Šādā veidā tiek izveidoti ieteikumi un paziņojumi par individualizētām diagnostikas un ārstēšanas metodēm.

Jin un Kim [33] prezentē semantisku modeli, kas ir balstīts uz IETF YANG datu formāta. Autoru semantiskais modelis ir veidots tā, lai iekļautu YANG modelēšanas valodas terminoloģiju. Šis modelis dod iespēju sadarboties ar vairākām komponentēm savā starpā, piemēram, klientiem, serveriem un e-veselības ierīcēm. Autori prototipa ieviešanai izmanto gan sensorus, kas nosaka ķermeņa temperatūru, gan asins spiedienu, gan arī elektromiogrāfijas sensorus. Speciāli priekš sensoru referencēšanas modelis sevī ietver arī OCF terminoloģiju. Autoru izveidotā shēma ne tikai nodrošina vairāku ierīču datu apmaiņu savā starpā, bet arī palīdz izteikt sensora datus lietotājam saprotamā veidā. Autori ir izveidojuši arī e-veselības datu ontoloģiju, kas definēta YANG formātā un atbalsta dažādus datu formātu stilus.

3.2. Ontoloģiju kvalitātes novērtēšana

Ontoloģiju novērtēšana ir process ar kura palīdzību tiek novērtēta ontoloģijas kvalitāte un pareizība attiecībā pret dažādiem novērtēšanas kritērijiem. Novērtēšanas kritēriji var atšķirties atkarībā no tā, kāda tieši ontoloģija tiek novērtēta un kādam nolūkam tas tiek darīts.

Dažādā ontoloģiju kvalitātes novērtēšanas literatūrā visbiežāk var sastapt tādu novērtēšanas metodi, kas atbilst vai ir atvasināta no četrām klasiskajām novērtēšanas metodēm. Šīs pieejas tiek izcili apkopotas šeit [13]. Viena no tām ir manuālā pieeja, kad ontoloģiju novērtē ekspertu grupa. Šādas pieejas ietvaros tiek spriests par to, cik labi ontoloģija atbilst dažādiem kritērijiem un prasībām [39]. Cita pieeja ir ontoloģijas salīdzināšana pret zelta standartu. [40] ir viens no piemēriem, kur šāda pieeja tiek izmantota. Vēl viena pieeja izmanto kādu ontoloģiju novērtēšanas lietojumprogrammu, lai vēlāk tās rezultātus varētu izmantot, lai spriestu par ontoloģijas kvalitāti. Visbeidzot eksistē ceturta pieeja, kurā tiek salīdzināta ontoloģija pret datu avotu. Viens no piemēriem šādai pieejas metodoloģijai ir atrodams šeit [14].

Balstoties uz Amith et al. [7], viena no izplatītākajām ontoloģiju novērtēšanas pieejām ir koncentrēties nevis uz pilnīgu ontoloģijas novērtēšanu, bet vērtēt dažādus ontoloģijas līmeņus vai aspektus atsevišķi. Dažādās publikācijās par ontoloģiju novērtēšanu šis līmeņu vai aspektu definīcijas mēdz atšķirties.

Balstoties uz Amith et al. [7], ontoloģijas novērtēšana ir vajadzīga, lai spētu noteikt ontoloģijas piemērotību datu apmaiņai un, balstoties uz Vrandecic [33], lai saglabātu atbilstību sekojošiem kritērijiem:

- 1) Pielāgošanās spēja
- 2) Skaidrība
- 3) Pilnīgums
- 4) Aprēķinu Efektivitāte
- 5) Lakonisms
- 6) Konsekvence
- 7) Organizatoriskā piemērotība
- 8) Precizitāte

Vairāki autori [7, 24, 33, 60] apgalvo, ka divi no svarīgākajiem uzdevumiem ontoloģiju novērtēšanā ir ontoloģiju verifikācija un validācija. Ontoloģiju verifikācija ir uzdevums, kura ietvaros tiek nodrošināts, ka ontoloģija tiek veidota pareizi un balstoties uz prasībām, bet validācijā tiek nodrošināts, ka ontoloģijas definīciju nozīme tiešām atbilst reālai pasaulei. Amith et al. [7] apgalvo, ka verifikācija pēta iekšējos, bet validācija ārējos aspektus.

Turklāt ontoloģiju (īpaši biomedicīnas ontoloģiju) novērtēšana dažādos literatūras rakstos mēdz dēvēt arī par kvalitātes nodrošināšanu. Dažreiz, kad ontoloģiju novērtēšanu veic bez to izstrādātāju klātbūtnes, to mēdz saukt arī par auditu [7].

3.3. Ontoloģiju novērtēšanas veidi

Kā jau tika minēts, vairāki autori [7, 24, 33] ir rakstījuši par ontoloģiju novērtēšanas veidiem, pirmajā plānā izceļot divas novērtēšanas kategorijas – verificēšanu un validēšanu. Taču ir pieejami citi veidi, kā līdz šim literatūrā ir tikušas novērtētas ontoloģijas. Balstoties uz Amith et al. [7], vieni no visietekmīgākajiem un visbiežāk citējamākajiem ir bijuši [12, 20, 46, 58]. Šajos darbos minētie ontoloģiju novērtēšanas veidi ir apkopoti 3.1. tabulā.

Duque-Ramos et al. [20] savā darbā raksta par OQuaRE kvalitātes modeli. OQuaRE mērķis ir definēt visus iespējamus elementus ontoloģijas novērtēšanai, kas ietver sevī novērtēšanas atbalstu, novērtēšanas procesu un metriku. OQuaRE var tikt pielāgots katrai konkrētai domēna

vajadzībai, vienkārši izvēloties sev nepieciešamos raksturojumus un raksturlielumus. Šim modelim pamatā ir pieci raksturlielumi un pirmais no tiem ir strukturālais. Tā sastāvā ir formālas un semantiski svarīgas ontoloģiskās īpašības, kuras tiek izmantotas vismodernākajās ontoloģiju vērtēšanas metodēs. Daži piemēri šīm īpašībām ir formalizācija, dublēšana, konsekvence, taču eksistē arī citas. Otrais no pieciem raksturlielumiem ir funkcionālā atbilstība. Ar to vērtē ontoloģijas atbilstību paredzētajam mērķim. Tiek vērtēta ontoloģijas shēmu un vērtību saskaņošana, zināšanu iegūšana, indeksēšana un saistīšana, un citas vērtības. Savukārt trešais raksturlielums ir ontoloģijas uzticamība. Autori kā piemērus min atjaunojamību un pieejamību. Šo raksturlielumu mērķis ir izmērīt ontoloģijas spēju uzturēt savu veikspējas līmeni noteiktos apstākļos. Visbeidzot seko tādi ontoloģijas raksturlielumi kā darbaspēja un apkope. Darbspēju autori raksturo kā piepūli, kas nepieciešama, lai izmantotu ontoloģiju. Galvenā apakš pazīme šim raksturlielumam ir mācāmība. Turpretim apkope ir domāta kā iespējamība ontoloģijai tikt modificētai tās vidē, prasībās vai specifikācijā. Kā galvenās apakš pazīmes šeit tiek minētas pārbaudāmība, maināmība un atkārtota izmantošana.

Tartir et al. [58] savā darbā apraksta 3 dažādas pieejas ontoloģiju novērtēšanai un validēšanai. Pirmā pieeja ir novērtēšana uz evolūcijas pamata. Kā raksta autori, laika gaitā ontoloģijas mainās un tas notiek pateicoties 3 dažādiem faktoriem: izmaiņām domēnā, konceptualizācijā vai specifikācijā. Domēna izmaiņas ir vispopulārākās un bieži notiek jaunu zināšanu pievienošanas dēļ. Izmaiņas konceptualizācijā var notikt mainoša pasaules redzējuma vai perspektīvas dēļ. Balstoties uz autoru rakstīto, universitāte atšķiras, ja uz to skatāmies no fakultātes vai studenta perspektīvas. Tāpēc, ja perspektīva mainās var izmainīties arī ontoloģija. Izmaiņas specifikācijā var notikt, kad ontoloģija tiek tulkota no vienas valodas uz citu. Kā autori min, viena valoda no citas atšķirsies ne tikai sintaksē, bet arī semantikā un izteiksmē. Otrā pieeja ontoloģiju novērtēšanai ir uz likumiem balstīta. Šī pieeja izmanto gan ontoloģijā eksistējošus likumus, gan arī lietotāju sniegtos likumus, lai spētu identificēt loģiskās kļūdas. OWL ontoloģijās, piemēram, varētu būt tādi gadījumi, kad objektiem vajadzētu būt atšķirīgiem vai, tieši otrādi, vienādiem, taču tas tā nav, vai arī to nav iespējams identificēt. Šādos gadījumos rodas likumu konflikts un ir atklāta kaut kāda kļūda. Šāda iespēja validēt ontoloģiju kvalitāti ir tikusi izveidota dažādās lietojumprogrammās. Daži piemēri ir RuleML lietojumprogramma vai arī Swoop modelis. Trešā pieeja izmanto statistiku par ontoloģijā eksistējošām zināšanām, lai spētu atklāt dažādas kvalitātes problēmas. Viens no visizplatītākajiem piemēriem ir OntoMetric metodoloģija. Tajā lietotājiem ir jāaizpilda vairākus mainīgos, kas raksturo ontoloģiju un, balstoties uz šo informāciju, ir iespējams identificēt vai

sistēmas prasības ir atbilstošas dotajai ontoloģijai. Dažas citas populāras tehnikas un programmas šajā pieejā ir AKTiveRank, ODEval, OntoClean kā arī pašu autoru [58] veidotais OntoQA. OntoQA pieeja izmanto piecas shēmas kvalitātes pazīmes kopā ar deviņām gadījumu kvalitātes pazīmēm, lai spētu izveidot ontoloģijas kvalitātes novērtējumu.

Obrst et al. [46] savā darbā detalizēti apraksta sešus dažādus veidus ontoloģiju kvalitātes novērtēšanai. Pirmā ontoloģiju novērtēšanas tehnika ir uz uzdevumiem balstīta. Šī ir ļoti noderīga un efektīva tehnika, jo tā ļoti spējīgi var izmērīt praktiskos ontoloģijas aspektus. Piemēram, tā var izmērīt tādus aspektus kā sistēmas secināmo komponentu atbilžu precizitāti, zināšanu bāzes mērogojamību, cilvēka spēju formulēt vaicājumus (šeit tiek izmantota vaicājumu valoda) un citus. Kā autori min, šādi novērojumi var palīdzēt raksturot zināšanu prasības. Šeit gan jāpiemin fakts, ka šāda, uz uzdevumiem balstīta, novērtēšana mēdz būt dārga un bieži vien to nevar otrreizēji izmantot kādai pilnīgi jauna uzdevuma novērtēšanas vajadzībai. Otra pieeja ir salīdzināšana ar domēna datu avotu. Šīs pieejas ietvaros ontoloģijas tiek salīdzinātas vainu pret citu ontoloģiju, vai kādu līdzīga domēna datu bāzi. Viens no šādas pieejas piemēriem ir gēnu ontoloģija, kas ir tikusi kartēta pret citām datu bāzēm gēnu domēnā. Tāpat ontoloģijas var tikt kartētas pret dažādiem ontoloģijas domēna dokumentiem. Trešā pieeja paredz cilvēku vērtēšanu pret dažādiem kritērijiem. Dažiem autoriem šī pieeja ir bijusi īpaši populāra biomedicīnas ontoloģiju vērtēšanā. Kā raksta autori, šīs pieejas principi ir visai saprātīgi un saprotoši. Svarīgākie novērtēšanas pietur punkti ir skaidra dokumentācija un konsekventi termini. Ceturtā un piektā pieeja ir dabiskās valodas apstrāde un realitātes izmantošana. Balstoties uz [7], šīs abas pieejas ir diezgan unikālas, jo līdz šim nav tikušas izmantotas nekur citur. Pirmajā no šīm abām pieejām tiek izmantota valodas vērtēšana, kas tiek realizēta atbildot uz jautājumiem (piemēram, izmantojot SPARQL valodu). Šādā veidā tiek mērīts atbildēto jautājumu apjoms. Runājot par uz realitātes balstītu novērtēšanas pieeju, tajā tiek noteikta semantiskā atbilstība starp ontoloģiju mezgliem un pārbaudītas izmaiņas, ko turpmākās versijas veikuši ontoloģijas kuratori.

Brank et al. [12] savā darbā aprakstīja vairākus ontoloģiju novērtēšanas veidus. Tāpat kā citi autori, arī autoru darbā tika veikta ontoloģiju novērtēšanas veidu klasifikācija. Tajā ir četri galvenie virzieni, kā notiek ontoloģiju novērtēšana. Tie ir 1) salīdzināšana pret zelta standartu, 2) salīdzināšana, izmantojot lietojumprogrammu un rezultātu analīze, 3) salīdzināšana ar datu avotu (piemēram, vairāku dokumentu izmantošana) un 4) tādi, kur novērtēšanu veic cilvēki. Tāpat autori piemin arī faktu, ka ontoloģijas pašas par sevi ir visai sarežģītas struktūras, kuru vienreizēja novērtēšana var nebūt tik vienkārša un nedot vēlamus rezultātus. Autori apgalvo, ka pastāv vairāki

iemesli ontoloģijas vērtēt pa līmeņiem, nevis visu vienā, un definē vairākus līmeņus, kuri būtu jāskatās. Pirmais līmenis ir leksikas, vārdu krājuma un datu līmenis, kura mērķis ir analizēt ontoloģijā iekļautos gadījumus un faktus un kādā veidā šie koncepti tiek identificēti. Viena no šī līmeņa tehnikām ir *edit distance*, jeb virkņu līdzības mērīšana. Otrs ir hierarhijas līmenis, kas koncentrējas uz hierarhiskām *is-a* attiecībām starp ontoloģijas jēdzieniem. Trešo līmeni autori sauc par citu semantisko attiecību līmeni. Tā kā iepriekšējā līmenī visas attiecības netiek atklātas, tad pārējās attiecības var tikt atklātas šajā līmenī un vērtētas atsevišķi. Nākamais ir konteksta vai lietojumprogrammas līmenis. Pastāv gadījumi, kad ontoloģija var sastāvēt no vairāku ontoloģiju kompleksa un līdz ar to vairākas definīcijas var eksistēt citās ontoloģijās un atsaukties viena uz otru. Tālāk seko sintaktiskais līmenis. Tā kā ontoloģijas tiek rakstītas kaut kādā formālā valodā, tad tām vienmēr ir jāatbilst sintaktiskām prasībām. Šis līmenis ir īpaši svarīgs tām ontoloģijām, kuras tika uzbūvētas manuālā veidā. Tāpat autori piemin struktūras, arhitektūras un dizaina līmeni. Arī šis līmenis ir svarīgs manuāli veidotām ontoloģijām, jo ir svarīgi, ka ontoloģija atbilst iepriekš definētiem arhitektūras un dizaina principiem.

3.1. tabula

Ontoloģiju novērtēšanas veidu kopsavilkums [7]

Tips	Veids	Brank et al. [12]	Obrst et al. [46]	Tartir et al. [58]	Duque-Ramos et al. [20]
Verifikācija	Leksiskas	X			
	Taksonomikas	X			
	Semantikas	X			
	Satura	X			
	Sintakses	X			
	Struktūras / Dizaina	X			
	Evolūcijas			X	
	Uz loģiku un noteikumiem balstīta			X	
	Uz metriku balstīta			X	
	Pareizības				X
Kvalitātes				X	
Validācija	Pret zelta standartu	X			
	Uz lietojumprogrammas balstīta	X	X		
	Uz datiem balstīta	X	X		
	Uz lietotāju pamatota	X	X		
	Uz NLP balstīts		X		
	Realitātes salīdzināšana		X		
	Kopienas sertifikācija		X		
	Uz metriku balstīta			X	
	Kategorizēšana				X
	Kvalitātes				X

3.4. Ontoloģijas un to novērtēšanas rīki

Pēdējo divu desmitgadu laikā ir parādījušies arvien vairāk pētījumu, kas cenšas aizpildīt ontoloģijas novērtēšanas jomā eksistējošu plaisu, izveidojot metodoloģiju, kas bieži vien pārtop par tiešsaistes rīku vai lietojumprogrammu. Šajā nodaļā tiks apskatīti šie rīki vai lietojumprogrammas. Svarīgi pieminēt, ka visi šie rīki visbiežāk pilda tikai vienu funkciju – tie vainu verificē vai validē ontoloģijas kvalitāti. Visi minētie rīki ir apkopoti 3.2. tabulā.

Tartir et al. [58] ir izstrādājuši OntoQA. Tas ir tīmeklī atrodamas ontoloģijas novērtēšanas rīks. Tas izmanto dažādas metrikas, lai mērītu ontoloģiju shēmas un zināšanu aspektus kā rezultātā ir spējīgs sarindot ontoloģijas pēc šīm dažādām metrikām. Ontoloģiju novērtēšana notiek divos veidos – tiek apskatītas shēmas metrikas un zināšanu bāzes metrikas. Shēmas metrikas novērtēšana koncentrējas uz ontoloģijas dizainu un sastāv no vairākām daļām. Pirmā daļa ir attiecību bagātīgums (angliski – richness), kas atspoguļo ontoloģijas attiecību veidu daudzveidību. Otrā shēmas metrikas sastāvdaļa pēta mantojuma bagātīgumu. Tās ietvaros tiek apskatīta gan vecāku klašu izplatība, gan arī aprakstīts informācijas sadalījums dažādos ontoloģiju mantojuma koka līmeņos. Trešā shēmas metrikas sastāvdaļa ir atribūtu bagātīgums. Šeit tiek apskatīts katrai klasei definēto atribūtu skaits. Zināšanu bāzes metrikas ietvaros tiek pētītas 5 dažādi ontoloģijas aspekti. Klases bagātīgums, kurā tiek pievērsta uzmanība tam kā gadījumi ir sadalīti pa klasēm. Klašu savienojamība, kurā tiek meklētas ontoloģijas galvenās klases, balstoties uz gadījumu attiecību grafiku. Klases nozīme, kuras ietvaros tiek rēķināta cik procenti no gadījumiem pieder klasēm mantojuma apakškokā, kuras sakņojas pašreizējā klasē. Ceturtā zināšanu bāzes metrikas sastāvdaļa ir kohēzija. Kohēzija ietvaros tiek meklētas dažādu gadījumu attiecības, lai noskaidrotu, kā divi gadījumi ir saistīti. Visbeidzot tiek pētīts attiecību bagātīgums, lai noskaidrotu to, cik daudz no shēmā klasei definētajām attiecībām tiek izmantotas gadījumu līmenī.

Lai arī mūsdienās eksistē dažādas metodoloģijas ontoloģiju veidošanai vai otrreizējai izmantošanai, Lozano-Tello un Gomez-Perez [38] apgalvo, ka nepastāv rīks, kas varētu lietotājam palīdzēt izlemt, kuru no vairākām ontoloģijām izvēlēties turpmākajam darbam vai projektam. Autori apgalvo, ka eksistē pieprasījums pēc rīka, kas, balstoties uz dažādām metrikām, spētu loģiski izsecināt, kuru ontoloģiju ir vērts izmantot un kura ontoloģija ir pilnībā neatbilstoša konkrētai sistēmai. Autori prezentē rīku ONTOMETRIC, kas palīdz lietotājam veikt šādu izvēli. ONTOMETRIC izmanto analītiskās hierarhijas procesu, lai palīdzētu ar kādu no sekojošām problēmām:

1. Izvēlēties visatbilstošāko ontoloģiju no vairāku ontoloģiju saraksta;
2. Saprastu ontoloģijas atbilstību kādam konkrētam projektam.

Ontoloģiju atbilstības novērtēšana sastāv no vairākiem soļiem. Pirmkārt, tiek nedefinēti projekta mērķi, priekš kura ontoloģija tiks izmantota. Šajā solī lietotājam ir jāizvēlas dažādu aspektu nozīme, piemēram, ontoloģijas terminu nozīmi, definīciju precizitāti, attiecību piemērotību un citu aspektu svarīguma noteikšanu. Nākamajā solī lietotājiem jāveido lēmumu koks, kurā jānorāda svarīgākās dimensijas ontoloģijas izvēlei. Trešajā solī, balstoties uz pirms tam jau izvēlētajiem kritērijiem, katrai brāļu mezglu kopai lietotājam jāizveido pāru salīdzināšanas matricas ar lēmumu koka kritērijiem. Tālāk notiek katras alternatīvās ontoloģijas īpašību vērtēšanas, izmantojot dažādas skalas. Visbeidzot tiek iegūts grafisks divu ontoloģiju piemērotības salīdzinājums, pēc kura var izvēlēties visatbilstošāko ontoloģiju [38]. Jāpiemin, ka darba rakstīšanas brīdī saite (kas norādīta šajā rakstā [38]) uz ONTOMETRIC rīku nedarbojās, kā arī autoru [38] minētā integrācija ar WebODE ontoloģijas inženierijas platformu [9] nav izveidojusies, jo WebODE platformas darbība ir pārtraukta.

Darba rakstīšanas brīdī autors nav spējis atrast strādājošu ONTOMETRIC [38] rīka saiti, taču viens no pirmajiem rezultātiem, kas parādās Google meklēšanā ir OntoMetrics [36] rīks. Līdzīgi kā ONTOMETRIC [38] arī OntoMetrics [36] izmanto metrisku aprēķinu, lai novērtētu ontoloģijas raksturlielumus un palīdzētu veikt izvēli starp vairākām līdzīgām ontoloģijām. Lantow [36] apgalvo, ka pēdējos gados ir uzrakstīti daudz pētījumu par ontoloģiju novērtēšanas rādītājiem, kas, piemēram, mēģina novērtēt ontoloģijas lasāmību, atkārtotas izmantošanas iespējas vai pielāgojamību. Taču, pēc autora domām, visi šie rīki vainu netiek vairāk uzturēti, vai arī nemaz nav izveidoti. Tāpēc autors ir izveidojis OntoMetrics, kas ir tiešsaistē pieejams dažādu ontoloģiju metriku kalkulēšanas rīks. OntoMetrics rīks novērtē ontoloģiju, balstoties uz divām dimensijām – ontoloģijas slāni un ontoloģijas jomu [36]. Ontoloģijas slānī OntoMetrics ir spējīgs novērtēt tādas aspektus kā populācija, konteksts, metadata, taksonomija un vārdu krājums. No ontoloģijas jomas puses, OntoMetrics savā vērtēšanā izmanto apjomīgu tvērumu ar dažādiem kritērijiem:

1. Domēnu tvērums – tiek novērtēts ontoloģijas pilnīgums, precizitāte, kodolīgums un konsekvence;
2. Konceptuālajā tvērumā – tiek novērtēta ontoloģijas skaidrība, lasāmība, skaitļošanas efektivitāte, pielāgojamība un citi aspekti;

3. Lietojumprogrammas tvērumā – tiek novērtēta ontoloģijas efektivitāte un vispārēja vērtība.

Amith et al. [6] raksta, ka pēdējos 10 gados ir parādījušies vairāki pētījumi saistībā ar ontoloģiju novērtēšanas rīku izveidi. Ņemot vērā to, ka eksistē milzīgs daudzums ar dažādām ontoloģiju kvalitātes novērtēšanas metodoloģijām, daudzi šīs jomas eksperti ir ķērušies klāt šo metodoloģiju iekļaušanai dažādos tiešsaistes un arī lietojumprogrammas risinājumos. Taču viens ir skaidrs, joprojām nav plaši pieejams vismaz viens standartizēts un ontoloģiju novērtēšanas jomā atdzīts rīks. Autori ir izveidojuši OntoKeeper, kas ir rīks, kuram vajadzētu nodrošināt ontoloģijas novērtēšanas funkcionalitāti zināšanu inženieriem ar labu lietojamību un vieglu pieejamību. OntoKeeper ir tiešsaistē pieejama lietojumprogramma. Tā ir veidota uz Java bāzes, spēj ielasīt OWL vai RDF failu formātus, un, izmantojot dažādas semiotiskās metrikas (kas balstās uz [15] pētījumu), spēj aprēķināt ontoloģiju aspektus, kas saistīti ar ontoloģijas kvalitāti. Rezultātā lietotājs par katru ontoloģijas semiotisko metriku iegūst skaitlisku rezultātu un īsu paskaidrojumu par rezultāta interpretēšanu. Turklāt, rīks ir spējīgs arī sniegt vispārēju kvalitātes aprēķinu, balstoties uz visiem esošajiem metriku rezultātiem.

Dividino et al. [19] ir izveidojuši S-OntoEval, kas ir uz semiotiku balstīts ontoloģiju novērtēšanas rīks. Rīks ir izveidots, lai palīdzētu lietotājiem novērtēt viņu ontoloģijas kvalitāti un atrastu vietas potenciālajiem uzlabojumiem. Rīks ir balstīts uz semiotisko teoriju [56] un izmantojot vairākus rādītājus ir spējīgs novērtēt ontoloģijas kvalitātes semantiskos, pragmatiskos kā arī sintaktiskos aspektus. S-OntoEval vērtēšanā ontoloģijas tiek sadalītas dažādos semiotikas līmeņos. Pirmkārt, tiek apskatīts strukturālais līmenis, kur notiek ontoloģijas sintakses novērtēšana kopā ar formālo semantikas vērtēšanu. Otrkārt, tiek vērtēts funkcionālais līmenis, kura ietvaros tiek analizēta ontoloģijas kognitīvā semantika. Visbeidzot, tiek apskatīts ar lietojamību saistītais līmenis, kurā novērtē ontoloģijas pragmatiku. Strukturālais līmenis strādā, izmantojot Pellet un RACER System argumentētājus. Tādā veidā tiek apskatīta loģiskā konsekvence (TBox un ABox) un grafu teorijas mēri. Funkcionālā līmenī tiek izmantota uz uzdevumiem balstītu vērtēšanas pieeja, bet lietojamības profilēšanas līmenī anotācijas apjoma kvantitatīva analīze.

Schober et al. [53] ir izveidojuši Protege pluginu – OntoCheck. Kā apgalvo autori, OntoCheck ļauj viegli pārbaudīt atbilstību ontoloģijas nosaukšanas konvencijām. Tāpat tas ļauj saprast par metadatu pilnīgumu, kā arī sniedz palīdzību ar konstatētajām kļūdām. Nosaukumu piešķiršanas konvencijas gadījumā, tas spēj atrast neskaidrus nosaukumus (piemēram, nosaukumus

ar 3 vai mazāk zīmēm, vai tieši otrādi, ļoti garus), kā arī pārbauda pieturzīmes, kardinalitātes rādītājus un citu. Turklāt tas var salīdzināt anotācijas rekvizītu un metadatu vērtības gan vienā klasē, gan arī starp vairākām dažādām klasēm. Turklāt OntoCheck var novērtēt etiķešu dublēšanos un nosaukumu sadursmes sinonīmu laukos dažādām klasēm. Tas savukārt palīdz izvairīties no slēptās dublēšanas. Rīkam eksistē arī skaitīšanas panelis ar kura palīdzību var redzēt ontoloģijas progressa uzraudzību lai novērtētu un veiktu dažādu analīzi. OntoCheck atrastās ontoloģiju kļūdas var labot, lai veicinātu konsekveni entītiņu nosaukumus, gan arī meta-anotāciju artefaktā.

OntoClean [27] ir metodoloģija ar kuru ir iespējams analizēt ontoloģijas kvalitāti. Balstoties uz Tiwari un Abraham [60], tas palīdz izmērīt nepiemērotas un nekonsekventas modelēšanas izvēles. Tā balstās uz filozofiskiem jēdzieniem, kā piemēram, būtība, identitāte un vienotība. OntoClean izmanto šos jēdzienus, lai aprakstītu izvēlētās ontoloģijas īpašību, klašu un attiecību paredzētās nozīmes aspektus. Šie aspekti, savukārt, veido formālas, no domēna neatkarīgas klašu īpašības, ko sauc arī par meta-īpašībām. Meta-īpašības uzliek vairākus ierobežojumus ontoloģijas taksonomiskajai struktūrai, un šādā veidā tiek nodrošināts ontoloģijas novērtējums. Balstoties uz Tiwari un Abraham [60], OntoClean ir WebODE plugins, taču balstoties uz [41] OntoClean metodoloģiju ir iespējams pielietot arī Protégé rīkā ar HermiT argumentētāju.

Horridge et al. [30] savā darbā nodemonstrē to, ka no vairāku ontoloģijas rīku klāsta, neviens nav spējis atklāt kādu no gramatiskās dublēšanas problēmām un tikai viens rīks (OilEd) ir spējis identificēt kļūdas jēdzienu taksonomijās. Tāpēc autori ir izveidojuši ODEval. ODEval ir ontoloģiju zināšanu reprezentēšana novērtēšanas rīks, kas ir valodas atkarīgs un strādā ar RDF, OWL un DAML+OIL, lai novērtētu jēdzienu taksonomijas. Autori apgalvo, ka šim rīkam vajadzētu palīdzēt izstrādāt ontoloģijas izvairoties no anomālijām dažādās ontoloģijas valodās. Balstoties uz Tiwari un Abraham [60], ODEval [30] ir WebODE plugins tāpat kā OntoClean.

Ontoloģiju novērtēšanas rīku kopsavilkums [60]

Ontoloģiju novērtēšanas rīks	Pieejamība	Vai ir brīvi pieejams bezmaksas?
OntoKeeper	Tīmekļa rīks	Tiešsaites meklēšanā nav brīvi atrodams (vai pieejams)
ONTOMETRIC	Tīmekļa rīks	Tiešsaites meklēšanā nav brīvi atrodams (vai pieejams)
OntoMetrics	Tīmekļa rīks	Jā
OOPs!	Tīmekļa rīks	Jā
Themis	Tīmekļa rīks	Jā
Protege	Ontoloģijas redaktors ar argumentētāju	Jā
TDDOnto	Protege plugins	Jā
ONTOQA	Tīmekļa rīks	Tiešsaites meklēšanā nav brīvi atrodams (vai pieejams)
S-OntoEval	Lietojumprogramma	Tiešsaites meklēšanā nav brīvi atrodams (vai pieejams)
OntoCheck	Protege plugins	Jā
OntoClean	WebODE plugins	Jā
ODEVAL	WebODE plugins	Tiešsaites meklēšanā nav brīvi atrodams (vai pieejams)
Moki	Tīmekļa rīks	Tiešsaites meklēšanā nav brīvi atrodams (vai pieejams)
OQuaRE	Tīmekļa rīks	Tiešsaites meklēšanā nav brīvi atrodams (vai pieejams)
XD-Tools	NeOn plugin	Jā

3.5. NCBO bioportāls

Mūsdienās eksistē tūkstošiem ontoloģiju, kas ir brīvi pieejamas internetā. Taču gandrīz katras ontoloģijas kvalitāte bieži vien ir stipri apšaubāma. Šis ir iemesls, kāpēc pēdējā laikā ir parādījušies dažādi pētījumi saistībā ar ontoloģiju veidošanas metodoloģijām kā arī ontoloģiju novērtēšanas metodoloģijām [10]. Balstoties uz Bandeira et al. [10], ir svarīgi ontoloģijas pārbaudīt pirms to lietošanas un tik pat svarīgi ir pārliecināties par šo kvalitāti uzreiz pēc ontoloģiju izveides.

Runājot par ontoloģiju atkārtotu izmantošanu ir sākumā jāsaprot, kur tās ontoloģijas vispār meklēt. Mūsdienās eksistē dažādas ontoloģiju krātuves, kurās tās var atrast, un no kurām tās var brīvi ņemt un izmantot. NCBO bioportāls [64] ir bieži ontoloģijas kopienā akceptēta biomedicīnas ontoloģiju krātuve [2, 7, 45, 52]. Tā ir viena no mūsdienās lielākajām biomedicīnas ontoloģiju krātuvēm, kurā ir brīvi pieejamas simtiem dažādu biomedicīnas ontoloģiju [7]. 2022. gada novembrī NCBO bioportāla mājaslapā bija pieejamas kopā 1,027 dažādas biomedicīnas ontoloģijas. Ņemot vērā to, ka 2018. gada janvārī tur kopā varēja atrast tikai 686 ontoloģijas, var secināt, ka to skaits joprojām turpina strauji augt [7]. NCBO bioportālā pieejamie ontoloģiju formāti ir ļoti plaši. Ontoloģijas tiek glabātas tādos formātos kā tīmekļa ontoloģijas valoda (OWL), Resursu apraksta ietvars (RDF) un citos. Turklāt, ontoloģiju dažādība NCBO bioportālā ir ļoti liela un aptver vairākas biomedicīnas tēmas. Tāpat ontoloģijas ir iespējams meklēt, vizualizēt, anotēt un pārlūkot, kas palīdz ar tām strādāt.

3.6. NCBO bioportāla ontoloģiju novērtēšanas kvalitātes pārskats

Ņemot vērā to, ka NCBO bioportāla platformā ir brīvi pieejamas simtiem dažādu ontoloģiju un konceptu, pēdējā laikā parādās aizvien vairāk pētījumu, kas cenšas noskaidrot šo ontoloģiju kvalitāti un atkārtotas izmantošanas iespējamību.

Ochs et al. [47] ir viens no pirmajiem pētījumiem, kas pētīja NCBO bioportāla ontoloģiju otrreizēju izmantošanu un rezultātā pieminēja arī otrreizējas izmantošanas ietekmi uz ontoloģiju kvalitāti. Darbā tika apskatītas 355 dažādas ontoloģijas, kas atrodas NCBO bioportālā. Autori apgalvo, ka no 355 ontoloģijām, 197 saturs ir otrreizēji izmantots no citas ontoloģijas. Turklāt tika atklāts, ka 108 ontoloģijas savu klašu veidošanai atkārtoti izmantoja kādas citas ontoloģijas klases. Savukārt 116 ontoloģijās ir izmantotas citu ontoloģiju īpašības klašu ierobežojumos. Interesanti, ka līdz šim vairākos pētījumos apskatītajā ontoloģiju otrreizējas izmantošanas metodē netika diskutēts par tās iespējamajām kvalitāti pasliktinošām sekām.

Ochs et al. [47] pētījumā tika atklāts, ka ontoloģiju otrreizējai izmantošanai ir iespējamās kvalitāti pasliktinošas sekas. Balstoties uz pētījuma rezultātiem, pētītajās ontoloģijās tika atrastas dublētās klases kā arī īpašības, kas pieder dažādām otrreizējas ontoloģijas versijām. Piemēram, tika atklāti gadījumi, kad vairākas ontoloģijas, kas aptver līdzīgu zināšanu jomu, tiek otrreizēji izmantotas un tāpēc modelētajā ontoloģijā parādās vairākas klases, kas reprezentē vienas un tās pašas entītijas. Citās ontoloģijās tika atrasti gadījumi, kad vienas un tās pašas ontoloģijas klases tiek izmantotas vairākās citās ontoloģijās, taču tās klases tiek ņemtas no dažādām importētās ontoloģijas versijām. Līdzīgi ir arī ar izmantotajām attiecībām. Vairākās ontoloģijās tika izmantotas citas ontoloģijas attiecībās, kuras ņemtas no dažādām versijām. Turklāt, dažās ontoloģijās ir uzreiz ieimportētas vairākas versijas no vienas un tās pašas ontoloģijas. Papildus, autori atklāja, ka dažās ontoloģijās ir ieimportēta kāda cita pilna ontoloģija, taču ne visas ieimportētās ontoloģijas entītijām tiek izmantotas. Šis savukārt rada jautājumu par to, kāpēc ir jāimportē veselas ontoloģijas, ja daļa no to zināšanām netiek izmantotas.

Līdzīgu darbu ir veikuši Halper et al. [29], kas pētīja ontoloģiju satura atkārtotas izmantošanas kvalitātes nodrošināšanu. Autori apgalvo, ka ontoloģiju satura atkārtota izmantošana ir ļoti svarīga mūsdienu ontoloģiju veidošanas sastāvdaļa. Kaut arī tas ir ļoti sarežģīts darbs, tas var samazināt darba apjomu, kas ir saistīts ar ontoloģiju veidošanu no jauna. Turklāt, tas var nodrošināt labāku saskaņošanu ar esošo zināšanu modelēšanu. Autori savā pētījumā apskatīja NCBO bioportālā pieejamās ontoloģijas 2015. gada aprīlī. Kopumā tika apskatītas 355 ontoloģijas,

no kurām vairāk kā puse (197 ontoloģijas) otrreizēji izmantoja kādas citas ontoloģijas saturu. Pētījuma rezultāti atklāja trīs galvenās problēmas, kas saistītas ar citu ontoloģiju zināšanu atkārtotu izmantošanu. Pirmā problēma ir klašu un īpašību dublikāti. Autoru pētījums parāda, ka no 149 ontoloģijām, kas otrreizēji izmantoja kādas citas ontoloģijas klasi, 30 procentiem bija izveidojies vismaz viena dublicētā klase. Runājot par īpašību otrreizēju izmantošanu, tika atklāts, ka no 31 ontoloģijas, 20 ontoloģijām bija izveidojušies īpašību dublikāti. Otra populārākā ontoloģiju problēma bija saistīta ar versiju izveidi. Visbiežāk šī problēma bija saistīta ar ontoloģijām, kas otrreizēji izmanto tieši BFO ontoloģijas zināšanas. Kopā tika identificēti 3 procenti no visām NCBO bioportāla ontoloģijām, kurām ir identificētas dažādas importēto ontoloģiju versijas. Trešā atklātā problēma bija saistīta ar importēto ontoloģiju trūkumu vai arī importēto ontoloģiju nederīgu vienoto resursu identifikatoru (URI).

Balstoties uz vairākiem pētījumiem [4, 5, 6, 7], ir skaidrs, ka starp NCBO bioportālā pieejamajām biomedicīnas ontoloģijām tikai mazai daļai ir pieejams nodokumentēts kvalitātes novērtēšanas apraksts. Lielākai daļai ontoloģiju kvalitātes novērtējums nav pieejams un, iespējams, tāds nekad nav arī veikts. Līdz ar to, biomedicīnas ontoloģiju ekspertu vidū pastāv vienošanās, ka biomedicīnas ontoloģijām visbiežāk pietrūkst dokumentēts formāls novērtējums. Tāpēc ir skaidrs, ka ir nepieciešama izpēte gan biomedicīnas ontoloģiju novērtēšanas jomā, gan arī ir skaidrs, ka trūkst novērtēšanas metodoloģijas standarts, pēc kā ontoloģiju veidotāji spētu vadīties.

Amith et al. [7] veica apjomīgu izpēti par NCBO bioportāla ontoloģiju kvalitāti. Rezultātā tika noskaidrots, ka tikai 15 ontoloģijām no 200 tika aprakstīta ontoloģijas novērtēšana to veidošanas dokumentos. Turklāt, autori noskaidroja, ka no 15 ontoloģijām ar eksistējošiem kvalitātes novērtēšanas aprakstiem, tikai puse izvērtēja informācijas vai formālās valodas pareizību. Turklāt, tikai pusei tika atrasti dokumentēti pierādījumi par holistiskā kvalitātes novērtējuma izmantošanu. Nevienai ontoloģijai netika atrasta uz reitingiem balstīta novērtējuma izmantošana.

3.7. Ontoloģiju kvalitātes nodrošināšanas pārskats

Amith et al. [7] apskatīja pēdējo gadu populārākos pētījumus par biomedicīnas ontoloģiju kvalitātes nodrošināšanas metodēm. Tika apskatīti pētījumi starp 2009. gadu un 2017. gadu. Šajos pētījumos aprakstītas kvalitātes nodrošināšanas metodes var klasificēt šādās pieejās:

- 1) Uz struktūru balstītas pieejas
- 2) Uz leksiku balstītas pieejas
- 3) Uz semantiku balstītas pieejas
- 4) Uz abstrakcijas tīklu balstītas pieejas
- 5) Lielo datu pieejas
- 6) Uz pūļa resursiem balstītas pieejas
- 7) Savstarpējās validācijas pieejas
- 8) Hibrīdās pieejas
- 9) Uz korpusu balstītas pieejas
- 10) Citu veidu pieejas

Taču Amith et al. [7] apgalvo, ka visas šīs pieejas koncentrējas uz satura pareizību. Visbiežāk tās ir dažādu klašu vai atribūtu neatbilstības un šo neatbilstību identificēšana. Tātad mērķis visbiežāk ir vienāds – samazināt dublikātus, uzlabot modelēšanas konsekvenci kā arī konceptu skaidrību. Taču par ontoloģijas satura atbilstību visbiežāk netiek domāts. Tiek pieņemts, ka labā kvalitāte pati par sevi nozīmē ontoloģijas atbilstību mērķim.

4. TERMINOLOĢIJA

4.1. Ontoloģijas

Ontoloģija ir aksiomu un entītiju kopums. Aksiomas ontoloģijā ir vismazākā vienība un tās definē attiecības starp ontoloģijas entītijām vai entītiju nosaukumiem. Aksiomas ir apgalvojumi, kas norāda to, kas šajā domēnā ir patiess un kas nav patiess. Ir dažādas aksiomas – fakti, anotācijas vai terminoloģiskas aksiomas. Terminoloģiskās aksiomas ir vainu klases vai īpašību aksiomas [11, 62].

Ontoloģija tiek veidota, izmantojot kādu ontoloģijas veidošanas valodu. Ontoloģijas valoda nosaka, kādas valodas konstrukcijas tiks lietotas un kāda būs formālā semantika, lai izveidotu ontoloģiju. Mūsdienās eksistē dažādas ontoloģiju valodas un tās dalās dažādās kategorijās. Viena no kategorijām ir tradicionālās ontoloģiju veidošanas valodas. Tās ir tādas valodas kā F-Logic vai Ontolingua. Cita kategorija ir valodas, kas izmanto iezīmēšanas shēmu, lai kodētu zināšanas. Šajā kategorijā ir tādas valodas kā OWL, RDF, RDFs un citas [11, 62].

Ontoloģijas sevī iekļauj dažāda veida zināšanas, kas ne vienmēr ir tikai terminoloģiskās zināšanas. Tās var būt terminu definīcijas datu aprakstīšanai, formālas attiecības starp terminiem kā arī vienkārši paši termini, kas apraksta gan pamat faktus, gan arī indivīdus. Tāpat ontoloģiju zināšanas var būt arī termini, kas apraksta indivīdu savstarpējās saiknes [46, 62].

Ontoloģijas parasti tiek glabātas kā fails. Šādu failu, kas ir ontoloģijas informācijas resurss sauc par ontoloģijas dokumentu un tas ir mašīnapstrādājams. Procesu, kurā izveidotā ontoloģija tiek pārveidota par failu sauc ontoloģijas serializāciju. Tīmekļu ontoloģiju var tikt serializētas dažādos standartos, piemēram, populārākie ir RDF, XML, OWL Abstract Syntax vai OWL Functional Syntax [62].

4.2. OWL

Tīmekļa ontoloģijas (Web ontologies) ir ontoloģijas, kuras tiek veidotas kādā no standarta tīmekļa ontoloģiju valodām. Tās ir tādas valodas kā RDF (Resource Description Framework), RDFs (Resource Description Framework schema) un OWL (Web Ontology Language) [62]. Tīmekļa ontoloģijas valoda, jeb OWL (dēvēta arī par “OWL 2”) ir semantiskā tīmekļa valoda. Tā ir jaunākais standarts ontoloģijas valodās un tā ir izveidota, lai atspoguļotu bagātīgas un sarežģītas zināšanas par lietām un šo lietu attiecībām [30].

4.3. Fakti

Par faktiem (mēdz saukt arī par individuālu aksiomu) ontoloģijās sauc atribūtus, saistības, instantiācijas, vienlīdzības un nevienlīdzības [11, 62].

Instantiācijām ir sekojoša specifikācija:

ClassAssertion (CE a), kur CE ir klases izteiksme (Class Expression) un a ir indivīds. Šī specifikācija norāda, ka indivīds a ir klases izteiksmes CE gadījums.

Pozitīva objekta īpašuma gadījumam ir sekojoša specifikācija:

PositivePropertyAssertion(OPE a b), kur OPE ir objekta īpašuma izteiksme un a, b ir indivīdi. Šī specifikācija norāda, ka a ir savienots ar b, izmantojot objekta īpašuma izteiksmi OPE.

Negatīva objekta īpašuma gadījumam ir sekojoša specifikācija:

NegativePropertyAssertion(OPE a b), kur OPE ir objekta īpašuma izteiksme un a, b ir indivīdi. Šī specifikācija norāda, ka a nav savienots ar b.

Datu tipa īpašuma gadījumam ir sekojoša specifikācija:

PropertyAssertion(R a v), kur R ir datu tipa īpašuma izteiksme, a ir indivīds un v ir literālis.

Līdzīgi kā iepriekš, eksistē arī negatīva specifikācija datu tipa īpašuma gadījumam:

NegativePropertyAssertion(R a v).

Svarīgi ir pieminēt arī individuālās vienlīdzības aksiomu:

SameIndividual(a₁ a₂ a₃... a_i). Tā nosaka, ka visi a_i, $1 \leq i \leq n$ attiecas uz vienu un to pašu indivīdu. Tāpat eksistē arī individuālā nevienlīdzības aksioma:

DifferentIndividual(a₁ a₂ a₃... a_i), kas nosaka pretējo – visi a_i, $1 \leq i \leq n$ nav viens un tas pats indivīds.

4.4. Aksiomas

Aksiomas strādā kā OWL pamats. Tās atspoguļo patiesības apgalvojumus semantiskajās zināšanās. Aksiomas saista kopā entītijas, kas ir domēna pamatjēdzieni un īpašību nosaukumi. Var būt gan klašu, gan īpašību aksiomas [30].

4.4.1. Klases aksiomas

Klases aksiomas iedalās 4 dažādās kategorijās – SubClassOf, EquivalentClasses, DisjointClasses, DisjointUnion [11, 62]. Pirmā kategorija ir subsumpcija ar sekojošu strukturālu specifikāciju [11, 62]:

SubClassOf(CE1 CE2). Šī aksioma norāda, ka klases izteiksme CE1 ir klases izteiksmes CE2 apakšklase. Piemēram, CE1 bērns būs apakšklase CE2 cilvēks. Šāda aksioma var apraksīt gan vienkāršas subsumpcijas, gan sarežģītas subsumpcijas, gan arī norādīt aprakstu [11, 62]. Vienkāršu subsumpciju gadījumā gan apakšklase, gan virsklase ir klašu nosaukumi. Taču sarežģītu subsumpciju gadījumā gan apakšklase, gan virsklase ir sarežģītas klases izteiksmes. Vērts pieminēt, ka sarežģītu subsumpciju gadījumos ontoloģiju lietotājiem var kļūt grūti saprast, kas tieši ir domāts ar šo subsumpciju, šādā veidā ierobežojot darbu ar ontoloģiju [62].

Aprakstu gadījumā viena klase ir klases nosaukums un otra ir sarežģītas klases izteiksme. Gan apakšklase, gan virsklase var būt gan viens, gan otrs.

Līdzvērtīgu klašu aksiomas forma ir sekojoša:

EquivalentClasses(CE1 ... CEn). Šī aksioma nosaka, ka klases izteiksmes CE_i ($1 \leq i \leq n$) ir semantiski vienādas viena otrai. Arī klašu izteiksmju gadījumā ir trīs dažādas kategorijas – klases ekvivalences, sarežģītas klases ekvivalences un definīcijas. Klases ekvivalences gadījumā abas klases ekvivalences aksiomas klases izteiksmes ir klašu nosaukumi. Turpretim sarežģītās klases ekvivalencē abas klases ir sarežģītas klases izteiksmes. [62] pievērš uzmanību tam, ka arī šajā gadījumā (līdzīgi kā ar subsumpcijām) šīs sarežģītās klases ekvivalences lietotājiem var būt ļoti grūti saprast. Tāpat, līdzīgi kā ar subsumpcijām, definīcijas sauc par tādām klases ekvivalencēm, kad viena klases ekvivalence ir klases nosaukums un otra ir sarežģīta klases izteiksme [11, 62].

Nesavienotu klašu aksiomai ir sekojoša forma:

DisjointClasses (CE1 ... CEn). Šī klases aksioma nosaka, ka visas klases izteiksmes CE_i ($1 \leq i \leq n$) ir nesavienotas savā starpā, tātad šīm klasēm nav kopīgu indivīdu.

Nesavienotu savienību klašu aksiomai ir sekojoša forma:

DisjointUnion(C CE1 ... CEn). Šī klases aksioma nosaka, ka klase C ir nesavienota savienība klases izteiksmēm CE_i ($1 \leq i \leq n$).

4.4.2. Īpašības aksiomas

Ontoloģijās īpašības aksiomas apraksta īpašību semantiku. Īpašības ir sarežģītākas par klasēm tādā ziņā, ka tās nevar aprakstīt, izmantojot tikai vienu aksiomu. Vienīgās īpašību izteiksmes ir apgrieztās īpašības un īpašību ķēdes. Īpašību aksiomas var izmantot četros dažādos veidos – lai definētu attiecības starp īpašībām, lai definētu īpašību aksiomu domēnu vai diapazonu, lai definētu īpašību aksiomu veidu, vai lai definētu atslēgas indivīdiem. Attiecības starp īpašībām ir apakšīpašības, līdzvērtīgas īpašības, nesaistītas īpašības un apgrieztās īpašības [62].

Ir pieejamas dažādas īpašību deklarācijas, lai deklarētu konkrētu īpašību veidus. Piemēram, funkcionālās, apgrieztās funkcionālās, refleksīvās, nerefleksīvās, simetriskas, asimetriskas un tranzitīvās īpašuma deklarācijas [62].

Īpašību aksiomām ir iespēja definēt atslēgas. Atslēgas ir domātas lai varētu identificēt indivīdu identitāti un apvienot datus no dažādiem avotiem [62].

4.5. Anotācijas

Anotācijas ontoloģijās izmanto anotāciju īpašību, lai spētu savienot elementu, piemēram, aksiomu, ontoloģiju vai entītiju, ar anotācijas vērtību. Anotācijas izmanto, lai vienkārši pievienotu vairāk informācijas par kādu no šiem elementiem. Piemēram, izmantojot anotāciju, ir iespējams pievienot informāciju par to kad tika ieviesta kāda klase vai kuras īpašības jau ir novecojušas [62].

Tāpat ar anotācijām ir iespējams pievienot arī vairāk informācijas par pašu ontoloģiju. Piemēram, pievienot ontoloģijas autoru, tās versiju vai tamlīdzīgi [62].

4.6. Entītijas

Ontoloģijās entītijas ir gan klases, gan datu tipi, gan personas, gan īpašības. Entītijas ir ontoloģiju pamatelementi un ar to palīdzību tiek definēti ontoloģiju termini [11, 62].

4.6.1. Indivīdu entītijas

Indivīdi ontoloģijās apzīmē faktiskos objektus, kuriem vainu ir skaidrs nosaukums (kas tiek izmantots, lai uz to indivīdu varētu atsaukties), vai arī nav dots nosaukums – tie ir anonīmi. Šādi anonīmi indivīdi ir bez globāla vārda un tāvad tie ir lokāli savai ontoloģijai. Šādus anonīmus

individus var identificēt tikai netieši, piemēram, izmantojot apgrieztās funkcionālās īpašības, atslēgas vai nominālvērtības [11, 62].

4.6.2. Klašu entītijas

Klases ir indivīdu kopas un tās tiek padotas kā klases izteiksmes. Var būt divas dažādas klases izteiksmes – klases nosaukums vai sarežģītas klases apraksts. Kamēr pirmais ir vienkārši nosaukums, otrais definē klase ar citu ontoloģijas entītiju palīdzību. Klases var izteikt ar bagātīgu konstrukciju daudzveidību – kopu operācijām vai ierobežojumiem. Kopu operācijas var būt krustojumi, savienības papildinājumi vai nomināli. Turpretim ierobežojumi var būt īpašuma eksistenciāli ierobežojumi, universāli ierobežojumi, nekvalificēta vai kvalificēta skaita ierobežojumi un sevis ierobežojumi [11, 62].

4.6.3. Īpašību entītijas

Īpašību entītijas var būt divu veidu – objektu īpašības vai datu īpašības. Pirmās savieno divus dažādus indivīdus, bet otrās savieno indivīdu ar datu vērtību [62].

Īpašību entītijas tiek izmantotas, lai veidotu īpašību izteiksmi [11]. Īpašību izteiksmes visbiežāk ir vienkārši nosaukums. Taču ir arī sarežģītas īpašību izteiksmes, piemēram, apgrieztās īpašības un īpašību ķēdes [11, 62].

Apgrieztās īpašības tiek izmantotas, kad subjekts ar objektu tiek apmainīti vietām īpašību instancē. Šādos gadījumos izmanto formu `InverseOf(OP)`. Piemēram, īpašības nosaukuma *vecāks* vietā var izmantot `InverseOf(bērn)` īpašības izteiksmi [62].

Īpašību ķēdes, kā jau pēc nosaukuma var nojaust, tiek izmantotas, lai savienotu vairākas īpašību izteiksmes vienā ķēdē. Piemēram, kāda cilvēka *mammas māsa* ir tās personas *tante* un šādu ķēdi var aprakstīt ar sekojošu formu `SubObjectPropertyOf(ObjectPropertyChain(a:hasMother a:hasSister) a:hasAunt)` [11].

4.6.4. Ontoloģiju entītijas

Eksistē divu veidu ontoloģijas – ar nosaukumu un bez. Ontoloģija ar nosaukumu nozīmē, ka ontoloģijai ir dots nosaukums un uz to var atsaukties, izmantojot šo doto nosaukumu. Turpretī ontoloģijas bez nosaukuma nozīmē, ka uz to nevarēs atsaukties, izmantojot tās nosaukumu, jo tāds nav izveidots. Bez nosaukuma ontoloģijas ir iespējams atpazīt pēc to lokācijas, ja tāda eksistē [11, 62].

4.7. Saistība

OWL ir uz skaitļošanas loģiku balstīta valoda. Valodas semantika tiek veidota no precīzām specififikācijām, kas nozīmē, ka visam ontoloģijā ir neapšaubāma un ļoti precīza nozīme. OWL ir iespējams arī definēt sekas, kas parasti nozīmē, ka kaut kas ir saistīts ar kaut ko citu (tas var būt arī sekas kaut kam citam). Rezultātā vienmēr ir iespējams sekot šai saistību saitei un pārliecināties, ka aksiomas seko noteiktām vēlamajām sekām, kā arī pārbaudīt, vai netiek radītas noteiktas nevēlamas sekas. Piemēram, ja ir definēts **Migrēna SubClassOf Neuroloģiska Slimība**, un **Neuroloģiska Slimība SubClassOf Slimība**, tad mums arī vajadzētu būt iespējai saprast to vai **Migrēna SubClassOf Slimība** ir taisnība. Līdz ar to, ja aksiomu var atrast ontoloģijā, tas nozīmē, ka aksioma ir saistīta ar šo ontoloģiju [30].

4.8. Argumentēšana

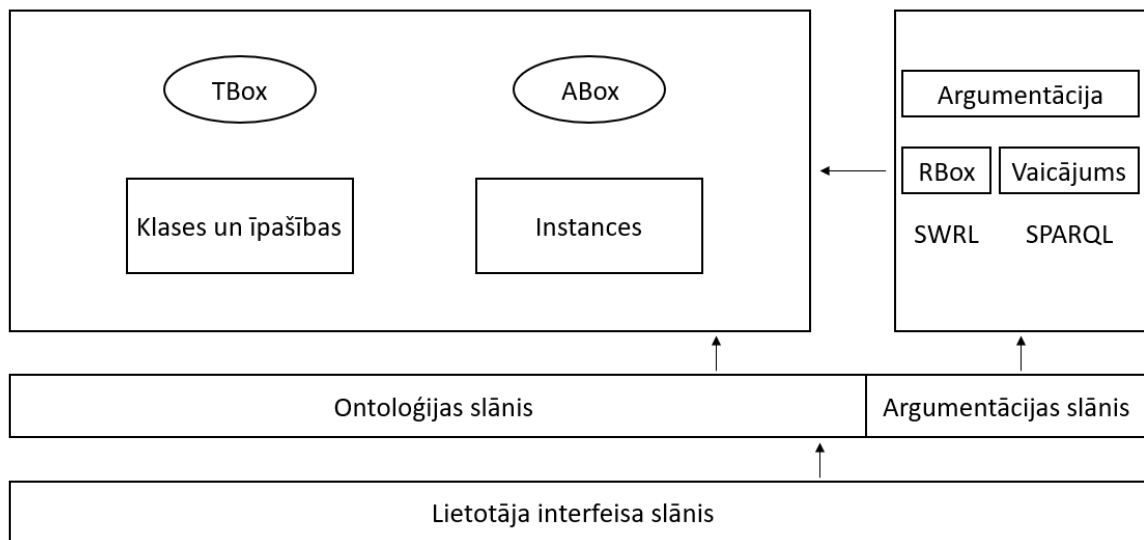
OWL dizains tika izveidots tā, lai tam būtu labvēlīgas skaitļošanas īpašības. Tas nozīmē, ka, izmantojot OWL, ir iespējams automatizēt seku aprēķināšanu, kam parasti izmanto argumentētāju. Semantiskie argumentētāji (saukt arī par spriešanas dzinējiem) ir programmatūras daļas, kas bieži ir iekļautas dažādās ontoloģijas projektēšanas programmatūrās. Tos var izmantot ontoloģijas izstrādes laikā vai lietojumprogrammas izpildes laikā. Semantiskos argumentētājus parasti izmanto, lai noskaidrotu to, vai ontoloģijā aprakstītais noved pie vēlamā rezultāta. Turklāt, ir ļoti svarīgi pārbaudīt arī citu saistību pusi. Ļoti bieži, ja ontoloģija ir pretrunīga, spriešanas laikā var pamanīt nevēlamas sekas (sauktas arī par loģikas kļūdām). Nekonsekvence starp ontoloģijām ir gan milzīga, gan arī bieži sastopama problēma [30]. Piemēram, [31] parāda polisēmijas izraisītu ontoloģisko nekonskvenci, kur precēta sieviete ir gan sieviete ar vīru, gan arī šķīrusies sieviete

(tiek uzskatīta par precētu, jo viņai agrāk ir bijis vīrs). Būtībā šādas ontoloģijas ir bezjēdzīgas, jo no tām nav iespējams izvilkt nekādu jēgpilnu secinājumu, jebkas var būt jebkas [30].

5. TEORĒTISKAIS IETVARŠ

5.1. Konceptuālais modelis

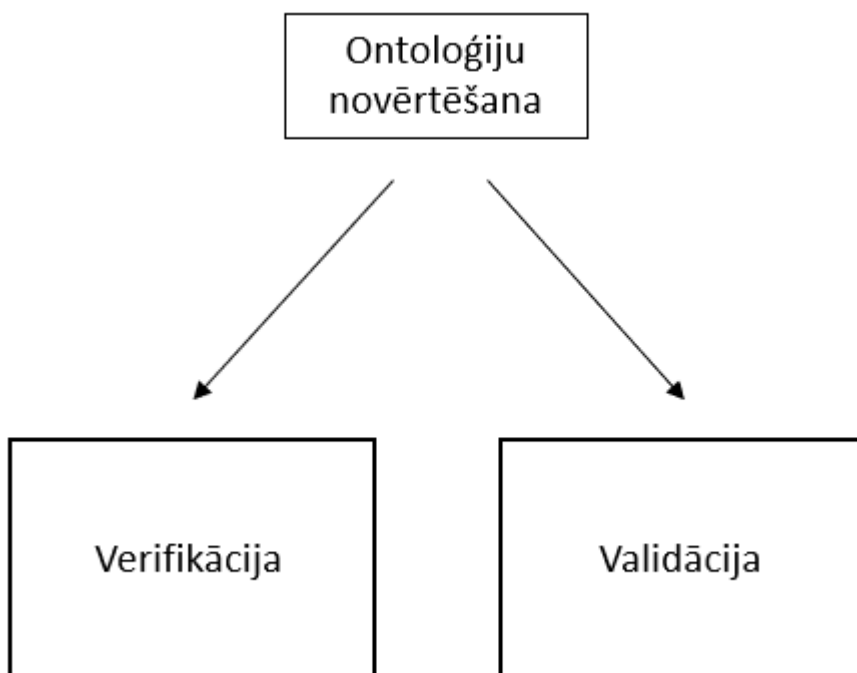
Konceptuālais modelis (5.1. att.), kas tiek izmantots šajā darbā ontoloģiju novērtēšanai, ir adaptēts no [60]. Tas sastāv no vairākiem slāņiem. Pirmais slānis satur sevī TBox un ABox. TBox apraksta viedo veselības jomu vispārīgo konceptuālo vārdu krājumu. ABox apraksta koncepcijas gadījumus un relāciju gadījumus, lai realizētu reālās pasaules objektus. Otrajā slānī atrodas argumentācijas slānis mijiedarbībai ar ontoloģiju, izmantojot SWRL noteikumus un SPARQL vaicājumus. Trešajā slānī atrodas lietotāja interfeisa slānis, tas palīdz piekļūt ontoloģijas un argumentācijas slānim.



5.1. att. Ontoloģiju kvalitātes novērtēšanas metodoloģijas ilustrācija [60]

Balstoties uz Sabou un Fernandez [50], ontoloģiju novērtēšana notiek divos dažādos veidos (5.2. att.). Pirmkārt, notiek ontoloģijas validācija. Tā ir ontoloģijas novērtēšanas darbība, kas salīdzina ontoloģijas definīciju nozīmi ar paredzēto pasaules modeli, ko izvēlēta ontoloģija mēģina konceptualizēt. Šajā darbībā mēs cenšamies atbildēt uz sekojošu jautājumu - vai mēs veidojam pareizo ontoloģiju? Otrkārt, notiek ontoloģijas verifikācija. Tā ir ontoloģijas novērtēšanas darbība, kas salīdzina ontoloģiju ar ontoloģijas specifikācijas dokumentu, proti, ontoloģijas prasībām un kompetences jautājumiem. Šādi tiek nodrošināts, ka ontoloģija ir veidota pareizi, tātad tā ir saskaņā

ar ontoloģijas specifikāciju. Šajā darbībā mēs mēģinām atbildēt uz jautājumu - vai mēs veidojam ontoloģiju pareizi?



5.2. att. Ontoloģiju novērtēšana [60]

Šajā darbā tiek izmantoti četri dažādi ontoloģiju novērtēšanas rīki – Themis, Protege, OOPs! un TDDOnto. Ontoloģiju prasību verifikācijai tiek izmantoti Themis un TDDOnto, bet zināšanu validācijas tika izmantoti Protege un OOPs!.

6. METODOLOĢIJA

Mūsdienu literatūrā eksistē vairāki piemēri, kad ontoloģijas tiek nepilnīgi novērtētas, izmantojot tikai vienu rīku [50, 60]. Kaut arī Amith et al. [7] piekrīt, ka “zelta standarts” ontoloģiju novērtēšanai un kvalitātes nodrošināšanai pašlaik neeksistē, citi autori [50, 60] ir pārliecināti, ka šobrīd viena no labākajām metodoloģijām ontoloģiju novērtēšanai sastāv no vairākiem rīkiem. Līdz ar to, šī darba metodoloģija ir adaptēta no Tiwari un Abraham [60] un izmanto vairākus ontoloģiju novērtēšanas rīkus.

6.1. Izvēlētās ontoloģijas

Mūsdienu veselības aprūpē ir sastopamas vairākas ontoloģijas, kas palīdz integrēt veselības aprūpes informāciju un ir savienojamas ar dažādiem veselības aprūpes rīkiem [60]. Tā kā informācija par visizmantotākajām ontoloģijām Latvijas veselības aprūpē autoram nebija pieejama, tika izvēlēts alternatīvs ceļš. Tika pētītas brīvi pieejamas ontoloģijas par veselības aprūpes datiem tādās vietnēs kā NCBO bioportāls [64] vai DBpedia Archivo [23].

NCBO bioportāls ir bieži ontoloģijas kopienā akceptēta biomedicīnas ontoloģiju krātuve [2, 7, 45, 52]. Pēdējos gados vairāki pētījumi ir veikuši ar ontoloģijām saistītu darbu, ņemot ontoloģijas tieši no NCBO bioportāla vietnes [4, 5, 6, 7]. Turklāt, ja salīdzina NCBO bioportālu ar DBpedia Archivo, tad lielāks biomedicīnas ontoloģiju daudzums, pēc autora domām, ir pieejams tieši NCBO bioportālā. Ņemot vērā augstāk minētos novērojumus un to, ka [64] ir pieejama iespēja kārot ontoloģijas pēc to popularitātes un apmeklējuma skaita, autors ontoloģiju meklēšanai izvēlējās NCBO bioportāla vietni. Ir vērts pieminēt, ka DBpedia Archivo ir iespēja redzēt ontoloģijas vērtējumu. Šis vērtējums (pie nosacījuma, ka ontoloģija ir ieguvusi visas četras zvaigznes) nozīmē to, ka ontoloģijai ir bijusi sekmīga konsekvences pārbaude, ko veic argumentētājs. Tas nozīmē to, ka lādējot ontoloģiju iekšā argumentētājā ir lielāka iespēja, ka tā tiks veiksmīgi ielādēta, salīdzinot ar citām ontoloģijām, kurām nav četru zvaigžņu reitings. Ņemot vērā to, ka NCBO bioportālā šāds vērtējums neeksistē, tas sarežģīt ontoloģiju izvēli un vēl vairāk pierāda vajadzību pēc pētījumiem saistītiem ar NCBO bioportāla ontoloģiju kvalitātes novērtēšanu.

NCBO bioportālā tika atrastas 10 dažādas biomedicīnas ontoloģijas (6.1. tabula). Ontoloģijas tika meklētas tā, lai iekļautu gan tādas, kurām ir liels klašu skaits, gan arī ontoloģijas ar mazāku klašu skaitu. Tika ņemts vērā arī to apmeklējumu skaits. Priekšroka tika dota tādām ontoloģijām,

kuru apmeklējumu skaits ir pēc iespējas lielāks. Visbeidzot, tika ņemts vērā arī ontoloģiju elementi. Priekšroka tika dota tādām ontoloģijām, kuru zināšanās eksistēja vismaz dažas attiecības un subsumpcijas starp klasēm.

Kad vēlamās ontoloģijas tika atrastas NCBO bioportālā, tad tās tika atrastas arī OBO Foundry [32], kur tika atrasts katras ontoloģijas IRI un tālāk izmantots, lai ar tām strādātu.

6.1. tabula

Šajā maģistra darbā izmantotās ontoloģijas

Nr.	Ontoloģijas nosaukums	Ontoloģijas IRI	Pieejama tiešsaitē	Klašu skaits	Apmeklējumu skaits NCBO BioPortal*
1	The BRENDA Tissue Ontology (BTO)	http://purl.obolibrary.org/obo/bto.owl	Jā	6569	1149 (Aug, 2022)
2	Clinical Measurement Ontology (CMO)	http://purl.obolibrary.org/obo/cmo.obo	Jā	3903	325 (Jun, 2022)
3	Ontology for Biomedical Investigations (OBI)	http://purl.obolibrary.org/obo/obi.owl	Jā	4673	528 (Feb, 2021)
4	Genotype Ontology (GENO)	http://purl.obolibrary.org/obo/geno.owl	Jā	424	60 (Aug, 2022)
5	Health Surveillance Ontology (HSO)	http://purl.obolibrary.org/obo/hso.owl	Jā	426	163 (Apr, 2022)
6	Human Disease Ontology (DOID)	http://purl.obolibrary.org/obo/doid.owl	Jā	18050	909 (Nov, 2021)
7	MFO Mental Disease Ontology (MFOMD)	http://purl.obolibrary.org/obo/MFOMD.owl	Jā	880	41 (Feb, 2022)
8	Evidence and Conclusion Ontology (ECO)	http://purl.obolibrary.org/obo/eco.owl	Jā	2095	122 (Sep, 2022)
9	Drug Interaction and Evidence Ontology (DIDEO)	http://purl.obolibrary.org/obo/dideo/release/2022-06-14/dideo.owl	Jā	617	183 (Apr, 2022)
10	Infectious Disease Ontology (IDO)	http://purl.obolibrary.org/obo/ido.owl	Jā	362	164 (May, 2022)

*Ņemot vīss vairāk apmeklētāko mēnesi

6.2. Ontoloģiju novērtēšana

Šajā darbā tika izveidoti 20 dažādi testi priekš trīs dažādiem rīkiem. Šie testi tika veidoti, balstoties uz [22] un [60] pētījumos veidotajiem testiem. Šie testi tiek izmantoti, lai novērtētu ontoloģijas ar tādiem rīkiem kā Themis, TDDOnto un Protege. Lai nonāktu pie standartizēta testu saraksta tika vispirms apskatītas visas ontoloģijas, lai novērtētu to saturu. Tika izlemts, ka gandrīz visas ontoloģijas sevī satur klases, īpašības, subsumpcijas starp klasēm, attiecības starp klasēm, nesavienotības (angliski – disjointness) kā arī vismaz dažus individuus un īpašības ar domēnu vai diapazonu. Līdz ar to tika izveidoti sekojoši 20 testi (6.2. tabula). T1-T3 testi veic vienkāršu pārbaudi, lai pārlicinātos, ka ontoloģija satur vajadzīgo klasi. T4-T5 līdzīgi pārbauda ontoloģijā esošās īpašības. T7-T9 pārbauda subsumpciju starp divām klasēm. T10-T12 pārbauda attiecības starp divām klasēm. T13-T15 veic pārbaudi klasēm ar nesavienotību. Tālāk T16-T18 pārbauda vai ontoloģijā eksistē minētie individū. T19 pārbauda domēnu starp īpašību un klasi. Visbeidzot, T20 pārbauda diapazonu starp īpašību un klasi. Dažās ontoloģijās nav definēti domēni, diapazoni vai neeksistē individū. Šādos gadījumos šie testi nav veikti konkrētām ontoloģijām. Balstoties uz Tiwari un Abraham [60], ar šo testu palīdzību ir iespējams novērtēt ontoloģiju pilnību, pareizību, skaidrību un konsekveni. Detalizēts testu saraksts katrai ontoloģijai ir atrodams 2. pielikumā.

Testa mērķi un testa izteiksmes sintakse [60]

Testa mērķis	Testa izteiksmes sintakse
T1 Ontoloģija satur klasi	[ClassA] <i>type</i> Class
T2 Ontoloģija satur klasi	[ClassA] <i>type</i> Class
T3 Ontoloģija satur klasi	[ClassA] <i>type</i> Class
T4 Ontoloģija satur īpašību	[PropertyP] <i>type</i> Property
T5 Ontoloģija satur īpašību	[PropertyP] <i>type</i> Property
T6 Ontoloģija satur īpašību	[PropertyP] <i>type</i> Property
T7 Starp divām klasēm ir subsumpcija	[ClassA] <i>SubClassOf</i> [ClassA]
T8 Starp divām klasēm ir subsumpcija	[ClassA] <i>SubClassOf</i> [ClassA]
T9 Starp divām klasēm ir subsumpcija	[ClassA] <i>SubClassOf</i> [ClassA]
T10 Attiecības starp divām klasēm	[ClassA] <i>SubClassOf</i> [PropertyP] <i>some</i> [ClassA]
T11 Attiecības starp divām klasēm	[ClassA] <i>SubClassOf</i> [PropertyP] <i>some</i> [ClassA]
T12 Attiecības starp divām klasēm	[ClassA] <i>SubClassOf</i> [PropertyP] <i>some</i> [ClassA]
T13 Nesavienotība	[ClassA] <i>disjointWith</i> [ClassA]
T14 Nesavienotība	[ClassA] <i>disjointWith</i> [ClassA]
T15 Nesavienotība	[ClassA] <i>disjointWith</i> [ClassA]
T16 Ontoloģija satur individu	[IndividualI] <i>type</i> [ClassA]
T17 Ontoloģija satur individu	[IndividualI] <i>type</i> [ClassA]
T18 Ontoloģija satur individu	[IndividualI] <i>type</i> [ClassA]
T19 Īpašībai ir domēns	[PropertyP] <i>domain</i> [ClassA]
T20 Īpašībai ir diapazons	[PropertyP] <i>range</i> [ClassB]

6.3. Darbā izmantotie ontoloģiju novērtēšanas rīki

6.3.1. Themis

Fernandez-Izquierdo un Garcia-Castro [22] ir izstrādājuši ontoloģiju novērtēšanas rīku – Themis. Themis ir tiešsaistes rīks, kurš nodrošina testa izteiksmju kopu, kuras pamatā ir leksika-sintaktiskie modeļi. Leksika-sintaktiskais modelis ir virkņu atbilstības modelis, kurš darbojas uz teksta marķieru un sintaktiskā struktūras pamata. Themis izmanto Leksika-sintaktiskais modeļus, lai saistītu dažāda veida prasības ar aksiomām ontoloģijā. Tā kā prasības ar aksiomām ir nepieciešams ieviest ontoloģijā, tad Themis izmanto šīs ieviešanas, lai veiktu pārbaudi. Rezultātā rīks spēj noteikt vai prasība ir izpildīta un aksiomas ir pareizi ieviestas, vai tomēr kaut kas trūkst.

Fernandez-Izquierdo un Garcia-Castro [22] uzskata, ka rīku Themis par noderīgu atdzīst gan domēnu ekspertu, gan arī paši ontoloģiju izstrādātāji. Piemēram, ontoloģiju izstrādātāji var izmantot Themis, lai pārbaudītu izstrādāto ontoloģiju pret tās funkcionālajām prasībām. Nefunkcionālās prasības Themis nav spējīgs pārbaudīt, jo šādas prasības nav iespējams formalizēt aksiomās.

Themis īstenotais testēšanas process ir izveidots un aprakstīts [21]. Tas ir testēšanas process, kas apraksta ontoloģijas validēšanu pret tai izveidotajām prasībām.

Testēšanas process tiek īstenots 3 soļos. Pirmais solis ir testa dizains. Šajā solī tiek izveidotas visas ontoloģijas zināšanu prasības. Šīs prasības tiek veidotas izmantojot *OWL Manchester Syntax* formātu un tiek noformētas kā testa izteiksmes. Ņemot vērā to, ka ontoloģijas URI netiek izmantoti, testa izteiksmes var tikt izmantotas vairākas reizes arī priekš citām ontoloģijām [22].

Otrais solis ir testa ieviešana. Šī soļa laikā lai pārbaudītu uzvedību katra testa izteiksme tiek formalizēta trīs dažādos veidos. Pirmajā veidā kā priekšnosacījums (kas ir SPARQL vaicājums). Te tiek pārbaudīts tas vai ontoloģijas testa izteiksmē pieminētie ontoloģijas termini ir vispār nodefinēti. Otrajā veidā tiek formalizēta kā palīgterminu deklarācijas kopa. Šeit, lai izpildītu apgalvojumus tiek pievienota pagaidu aksiomu kopa, kas deklarē palīg terminus. Visbeidzot testa izteiksme tiek formalizēta kā apgalvojumi. Apgalvojumi ir sagaidāmie rezultāti un aksiomu pāru kopums, kas domāti, lai parādītu dažādus scenārijus [22].

Trešais solis ir testa izpilde. Šis solis sastāv no trīs daļām un tā laikā tiek pārbaudīta ontoloģija attiecībā pret katru testa gadījumu. Pirmais solis izpilda uzrakstīto vai importēto vaicājumu, otrais solis pievieno palīg terminus deklarējošās aksiomas un trešais solis pievieno apgalvojumus. Katrs no šiem soļiem tiek saglabāti RDF formātā. Themis rīkā tiek izmantots Pellet15 argumentētājs kopā ar OWL API14. Tas savukārt ļauj šim rīkam ielādēt ontoloģijas un pārbaudīt to konsekveni. Vis beidzot, ir iespēja arī pārbaudīt vairākas ontoloģijas vienlaicīgi. Pēc vairāku ontoloģiju ielādes, katrs testa gadījums tiks izpildīts uz visām ontoloģijām, kas palīdz gan ātrāk, gan efektīvāk pamanīt kopīgās zināšanas starp vairākām ontoloģijām vienlaicīgi [22].

Balstoties uz Fernandez-Izquierdo un Garcia-Castro [22], pēc katra izpildīta testa Themis lietotājam parāda vienu no četriem rezultātiem:

1. *Undefined terms.* Šī Themis atbilde nozīmē to, ka izmantotie termini testa izteiksmē nav nodefinēti izvēlētajā ontoloģijā un testa gadījums nav izpildīts.
2. *Passed.* Šī Themis atbilde nozīmē to, ka testa izteiksmes priekšnosacījumi ir izturēti un testa izteiksmes rezultāti atbilst ontoloģijas modelētajām zināšanām.
3. *Absent relation.* Šī Themis atbilde nozīmē to, ka testa izteiksmes priekšnosacījumi ir izturēti, pretrunas nav atrastas, taču rezultāti nav atbilstoši gaidītajiem rezultātiem.
4. *Conflict.* Šī Themis atbilde nozīmē to, ka testa izteiksmes priekšnosacījumi ir izturēti, rezultāti nav atbilstoši gaidītajiem rezultātiem un tests noved pie konflikta.

Dažus no rezultātiem var apskatīt 6.1. attēlā.

<http://purl.obolibrary.org/obo/MFOMD.owl>
[See the glossary of terms](#)
[Remove](#)

2 **Check the tests**

You can load a test suite from a URL, adding an RDF file that uses the **VTC** ontology.

[Load from URI](#)

Or you can copy the test suite code, if the test suite uses the **VTC** ontology.

[Load from file](#)

Or you can add the tests directly. To add more than one test separate them by using ";".

The following link shows all the supported tests. In this other link you can also find some examples that can be useful to propose tests.

[Check](#)

Important information

- The test syntax is **case sensitive** (e.g., Sensor and sensor are detected as different terms)
- For generating the tests you have to use the **glossary of terms** (you can change it)
- The test syntax is available [here](#). You should insert the terms between brackets ("e.g., [Class] → Sensor"), while the italicised terms (e.g., type) are keywords that cannot be changed
- Terms in **red** indicate that they don't exist in the ontology, while the terms in **orange** indicate that the term exist in the ontology but the type (e.g., class or individual) is not the specified in the term
- The **Absent relation** result does not indicate that the test is not passed, but that something may be missing

Tests results

Test	Result	Problem	
BFO_0000062 domain BFO_0000003	Undefined terms	The terms in the test are not defined in the ontology	
BFO_0000054 range BFO_0000015	Absent relation	The ontology does not implement the requirement associated to the test	
IAO_0000227 type IAO_0000225	Passed	None	
IAO_0000124 type IAO_0000078	Passed	None	

6.1. att. Themis piemērs ar MFOMD ontoloģiju

6.3.2. TDDOnto

Balstoties uz Lawrynowicz un Keet [37], mūsdienu ontoloģiju novērtēšanas jomā trūkst tādi rīki, kas ir spējīgi veikt ontoloģiju verifikāciju. Visbiežāk modelētājs šādu verifikācijas soli neveic, tāpēc vienīgais kas atliek ir paļauties uz argumentētāja atbildi. Diemžēl mūsdienās neeksistē tāda viena ontoloģiju autorēšanas vieta, kas spētu sistemātiski pārbaudīt ontoloģijas. Tāpēc autori [37], ņemot piemēru no testu vadītas izstrādes programmatūras inženierijā, ir izstrādājuši TDDOnto – ontoloģiju novērtēšanas rīku. Izmantojot 82 dažādas ontoloģijas tika panākta rīka visefektīvākā ieviešanas stratēģija. Rīks tika izstrādāts tā, lai TBox spētu veikt klašu un īpašību TDD testus, bet ABox testētu instanču līmenī [60]. Kā secina autori [37], TDDOnto testi ir ātrāki par parastajiem argumentācijas testiem un līdz ar to tā ir mūsdienīga alternatīva līdz šim izmantotajai “mēģini un

skaties” (angliski – try and see) ontoloģiju novērtēšanas pieejai, kuras laikā tiek akli mēģināti argumentētāja testi un sagaidīts nezināms rezultāts.

TDDOnto testi tiek veikti Protege rīkā. Testiem ir nepieciešams ieslēgts argumentētājs un tie balstās uz SPARQL vaicājumiem. Pašlaik ir pieejamas divas dažādas versijas TDDOnto spraudnim. Šeit [8] ir pieejams sākotnējās versijas JAR fails, kuru var pievienot Protege spraudņu mapē. Šīs versijas lietošanas pamācību ar visiem skaidrojumiem var atrast šeit [35]. Otrās versijas TDDOnto spraudnis ir aprakstīts [34]. Taču JAR fails nekur nav atrodams priekš otrās versijas un pievienotajā *github* saitē nav iespējams atrast JAR failu, kā tas tiek minēts te [34].

Pirmajā TDDOnto versijā pēc katra izpildīta test TDDOnto lietotājiem piedāvā vairākus rezultātus [35]:

1. *Ok*. Šāda TDDOnto atbilde apzīmē atbilstību ontoloģijā modelētajām zināšanām.
2. *Undefined*. Šāda TDDOnto atbilde apzīmē zināšanu neesamību ontoloģijā.
3. *Failed*. Šāda TDDOnto atbilde apzīmē to, ka aksioma nav saistīta, vai arī to, ka aksioma nav atbilstoša.

Piemēru var apskatīt 6.2. attēlā.

Otrā TDDOnto versija ir uzlabota un atbildes ir detalizētākas. Pēc katra izpildīta test TDDOnto lietotājiem piedāvā četrus dažādus rezultātus [37, 60]:

1. *Absent*. Šī TDDOnto atbilde nozīmē to, ka aksioma ontoloģijā neeksistē, taču to var pievienot, neradot papildus neatbilstības.
2. *Inconsistent*. Šī TDDOnto atbilde nozīmē to, ka šo aksiomu neizdosies pievienot ontoloģijai, kā arī tā nav atbilstoša.
3. *Entailed*. Šī TDDOnto atbilde apzīmē atbilstību ontoloģijā modelētajām zināšanām.
4. *Incoherent*. Šī TDDOnto atbilde nozīmē to, ka testā izmantotā aksioma nevar tikt pievienota ontoloģijai, jo aksioma nav saistīta un ontoloģijā tās izveide nav sakarīga.

Šajā darbā tiek izmantota pirmā TDDOnto versija, kuras spraudnis ir brīvi pieejams [8].

The screenshot shows the TDDOnto web interface for the ontology 'ido' (http://purl.obolibrary.org/obo/ido/2017-11-03/ido.owl). The interface includes a navigation bar with 'Active ontology', 'Entities', 'Individuals by class', and 'DL Query'. The main content area is divided into two sections: 'Annotations' on the left and 'Tests' on the right. The 'Annotations' section lists various creators and contributors, including Alan Ruttenberg, Albert Goldfain, Alexander Diehl, Barry Smith, Bjoern Peters, Lindsay Cowell, and Jie Zheng. The 'Tests' section displays a list of tests with their results. A 'New test' input field is visible at the top right, containing the text 'BFO_000054 range BFO_000015'. Below the input field are buttons for 'Add test', 'Load tests', and 'Save tests'.

Test	Result
IDO_0000484 type Class	Undefined
IDO_0000469 type Class	Undefined
IDO_0000594 type Class	Undefined
RO_0001025 type Property	Undefined
RO_0002215 type Property	Undefined
RO_0000081 type Property	Undefined
BFO_0000144 subclassOf BFO_0000015	OK
IDO_0000466 subclassOf BFO_0000019	OK
IDO_0000433 subclassOf BFO_0000023	OK
IDO_0000482 subclassOf RO_0000052 some IDO_0000509	OK
IDO_0000635 subclassOf RO_0000052 some BFO_0000040	OK
IDO_0000584 subclassOf RO_0000091 some IDO_0000426	OK
IDO_0000425 disjointWith IDO_0000458	OK
IDO_0000438 disjointWith OGMS_0000031	OK
IDO_0000449 disjointWith IDO_0000580	OK
IAO_0000410 type IAO_0000409	Undefined
IAO_0000226 type IAO_0000226	Undefined
IAO_0000124 type IAO_0000078	Undefined
BFO_0000062 domain BFO_0000003	Undefined
BFO_0000054 range BFO_0000015	Undefined

6.2. att. TDDOnto piemērs ar IDO ontoloģiju

6.3.3. OOPs!

Balstoties uz Poveda-Villalon et al. [48], viens no populārākajiem veidiem ontoloģiju vērtēšanai sastāv no sekojošām daļām. Pirmkārt, tiek meklētas un pierakstītas populārākās kļūdas, kuras radās citiem ontoloģiju izstrādātājiem. Tālāk tiek veidota ontoloģija, balstoties uz iepriekš sarakstīto populārāko kļūdu sarakstu. Tiklīdz kāda kļūda tiek atrasta veidotajā ontoloģijā, tā tiek salabota. Autori [48] ir izveidojuši tiešsaistes ontoloģiju novērtēšanas rīku, kas sevī iekļauj iepriekš minēto ontoloģiju novērtēšanas metodi. Šo rīku sauc “*OOPs!*”.

Poveda-Villalon et al. [48] ar “*OOPs!*” izveidošanu ir snieguši divu veidu piensumu ontoloģiju novērtēšanas jomai. Pirmkārt, lai izstrādātu rīku, kura novērtēšanas metode balstās uz ontoloģiju veidošanā biežāk sastopamākajām kļūdām bija vajadzīgs šīs kļūdas identificēt un sagrupēt. Tāpēc autori ir izveidojuši milzīgu katalogu ar dažādām ontoloģiju veidošanā sastopamajām kļūdām. Šīs kļūdas tika sagrupētas trīs dažādās dimensijās – strukturālās kļūdas, funkcionālās kļūdas un lietojamības profilēšanas (angļu: usability profiling) kļūdas. Ņemot vērā to, ka visas kļūdas nav vienādi nozīmīgas, tās ir papildus sagrupētas trīs dažādos svarīguma līmeņos - kritiskas, svarīgas un nelielas nozīmes. Balstoties uz [61], šīs *OOPs!* kļūdas apzīmē sekojošo:

- *Kritiskas kļūdas*: Nozīmē to, ka tās ir obligāti jānovērš, citādi šādas kļūdas var ietekmēt gan argumentētāja darbību, gan arī ontoloģijas konsekvenci.

- *Svarīgas kļūdas*: Nozīmē to, ka tās nav kritiskas un tās nav obligāti uzreiz jāsteidzas novērst. Taču šādas kļūdas var ietekmēt ontoloģijas modelēšanu un to novēršana var uzlabot saprašanu par ontoloģijā modelētajām zināšanām.
- *Nelielas nozīmes kļūdas*: Varētu drīzāk saukt par atrastajiem ontoloģijas uzlabojumiem nevis kļūdām. Šādu uzlabojumu veicot, ontoloģija kļūs autentiskāka.

Autoru [48] kļūdu katalogs sevī ietver divus dažādus kļūdu avotus. Pirmkārt tas balstās uz iepriekšējiem ontoloģiju kļūdu modelējušiem darbiem. Otrkārt, tas ir papildināts ar vairāk nekā 693 ontoloģiju empīrisku analīzi. Rezultātā ir prezentētas 40 dažādas kļūdas. Taču uz šī darba rakstīšanas brīdi *OOPs!* kļūdu katalogs ir papildināts un veido sarakstu ar 41 kļūdām [49]. Gadījumā, ja atklātā kļūda neeksistē *OOPs!* - kataloga papildināšana ir iespējama, aizpildot speciālu formu, kas pieejama [49].

Balsoties uz [60, 61], *OOPs!* spēj pārbaudīt sekojošus ontoloģijas kritērijus - pilnīgumu, skaidrību, pareizību, konsekveni un lakoniskumu. Tiwari un Abraham [60] palīdz izprast, kas tiek analizēts zem katra no šiem kritērijiem:

- *Pilnīgums*: Pilnīgums tiek pārbaudīts divos dažādos līmeņos. Pirmkārt, tas tiek pārbaudīts shēmas līmenī. Šajā līmenī datu kopa tiktu uzskatīta par pabeigtu tikai gadījumā, ja to veido visi atribūti, kas nepieciešami uzdevuma veikšanai. Otrkārt, tas tiek pārbaudīts instances līmenī. Šeit savukārt datu kopa tiktu izskatīta par pabeigtu tikai gadījumā, ja to veido visi objekti, kas nepieciešami uzdevumu veikšanai.
- *Kodolīgums*: Kodolīgums tiek pārbaudīts apskatot datu kopas shēmas līmenī. Ja datu kopa nesatur liekus atribūtus, tad to var uzskatīt par kodolīgu. Līdzīgi arī instanču līmenī – jo mazāk lieku objektu, jo kodolīgāk.
- *Skaidrība*: Šeit tiek aplūkots tas, cik skaidri ir terminu apraksti. Terminu nosaukumiem ir jābūt saprotamiem un tie nedrīkst būt populāri. Turklāt, ontoloģijai nevajadzētu dot priekšroku klases aprakstam, ja eksistē definīcija. Jo skaidrāka ir ontoloģija, jo labāk tā interpretē aprakstīto terminu paredzamo nozīmi.
- *Pareizība*: Pareizība tiek pārbaudīta, apskatot to, cik pareizi ontoloģija attēlo reālās pasaules aspektus. Šeit ir svarīgi tas, cik jēgpilni ir klašu, attiecību un indivīdu apraksti.
- *Konsekvence*: Konsekvence tiek pārbaudīta caur pretrunām, kas eksistē ontoloģijā. Ja ontoloģijas datu kopā neeksistē pretrunīga informācija, tad šāda datu kopa tiks uzskatīta par konsekventu.

6.3. attēlā var apskatīt piemēru ar DIDEO ontoloģijas pārbaudi un vairākām kļūdām.

Scanner by URI: Scanner by URI
 Example:

Scanner by direct input:

Uncheck this checkbox if you don't want us to keep a copy of your ontology. [Go to advanced evaluation](#)

Evaluation results

It is obvious that not all the pitfalls are equally important; their impact in the ontology will depend on multiple factors. For this reason, each pitfall has an importance level attached indicating how important it is. We have identified three levels:

- **Critical** 🚫 : It is crucial to correct the pitfall. Otherwise, it could affect the ontology consistency, reasoning, applicability, etc.
- **Important** ⚠️ : Though not critical for ontology function, it is important to correct this type of pitfall.
- **Minor** 🟡 : It is not really a problem, but by correcting it we will make the ontology nicer.

[Expand All] | [Collapse All]

Results for P04: Creating unconnected ontology elements.	1 case Minor 🟡
Results for P08: Missing annotations.	6 cases Minor 🟡
Results for P11: Missing domain or range in properties.	81 cases Important ⚠️
Results for P13: Inverse relationships not explicitly declared.	58 cases Minor 🟡
Results for P19: Defining multiple domains or ranges in properties.	1 case Critical 🚫
Results for P20: Misusing ontology annotations.	4 cases Minor 🟡
Results for P22: Using different naming conventions in the ontology.	ontology* Minor 🟡
Results for P29: Defining wrong transitive relationships.	1 case Critical 🚫
Results for P30: Equivalent classes not explicitly declared.	2 cases Important ⚠️
Results for P32: Several classes with the same label.	5 cases Minor 🟡
Results for P34: Untyped class.	5 cases Important ⚠️
Results for P36: URI contains file extension.	ontology* Minor 🟡
Results for P39: Ambiguous namespace.	ontology* Critical 🚫
Results for P41: No license declared.	ontology* Important ⚠️
SUGGESTION: symmetric or transitive object properties.	3 cases

Want to help?

- Suggest new pitfalls
- Provide feedback

Documentation:

- Pitfall catalogue
- User guide
- Technical report

Related papers:

- IJSWIS 2014
- EKAW 2012
- ESWC 2012 Demo
- Ontoqual 2010
- CAEPIA 2009

Web services:

- REST Web Service

Developed by:



6.3. att. OOPs! piemērs ar DIDEO ontoloģiju

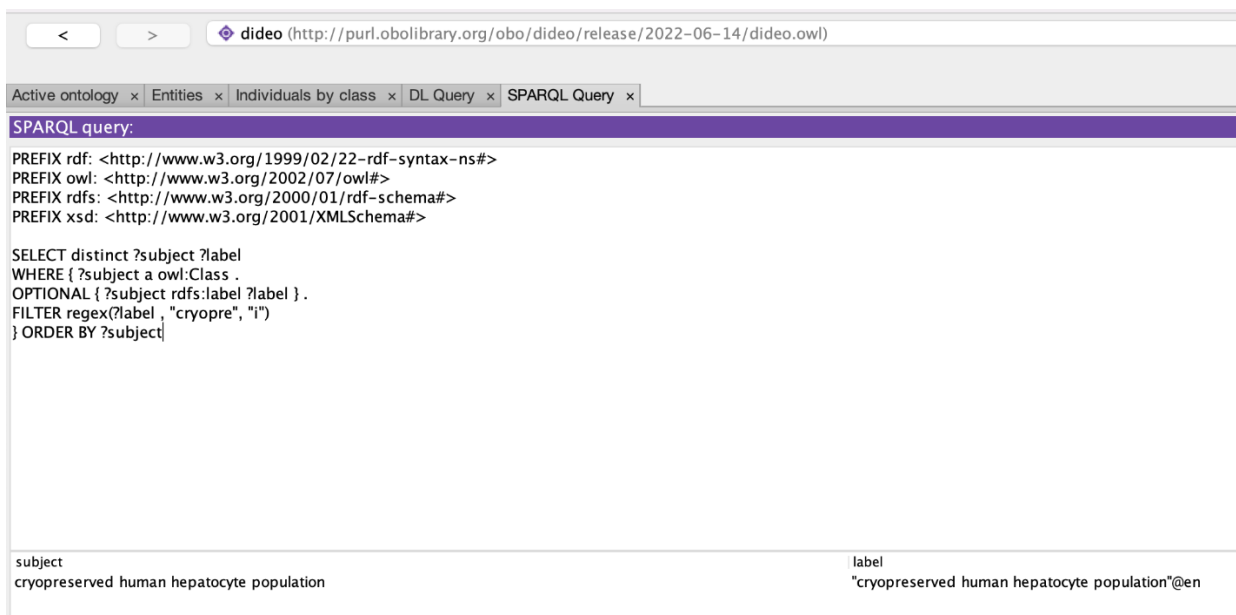
6.3.4. Protege

Protege ir bezmaksas atvērta koda ontoloģiju redaktors kā arī ietvars viedo sistēmu izveidei [43]. Protege ļauj validēt ontoloģijas, izmantojot iebūvēto argumentētāju. Pēc noklusējuma, lejupielādētai Protege lietotnei (Windows versijai) bija pieejams HermiT 1.4.3.456 argumentētājs. Macbook versijai bija pieejami vairāki argumentētāji pēc noklusējuma, taču tika lietots tāds pats kā uz Windows versijas - HermiT 1.4.3.456.

HermiT ir argumentētājs, kas ir domāts ontoloģijām uz OWL valodas pamata. Tas izmanto padoto ontoloģijas OWL failu un spēj noteikt to, cik konkrētā ontoloģija ir konsekventa. Tāpat HermiT spēj identificēt subsumpciju attiecības starp klasēm un veikt dažādus citus testus [17].

Hermit pamatā ir “hypertableau” algoritms un tas ir pirmais publiski pieejamais argumentētājs, kas izmanto šādu algoritmu priekš OWL ontoloģijām. Hermit argumentētājs izmanto tiešo semantiku un tas ir izturējis visus OWL 2 atbilstības testus [17].

Iekš Protege ontoloģiju validācija ir iespējama vairākos veidos. Kā norāda Tiwari un Abraham [60], ontoloģiju testa gadījumus ir iespējams pārbaudīt caur Protege meklētāju. Tāpat ir iespējams testa gadījumus noformēt SPARQL vaicājumu veidā un izpildīt ar ieslēgtu argumentētāju (6.4. attēls).



The screenshot shows a web browser window with the URL `http://purl.obolibrary.org/obo/dideo/release/2022-06-14/dideo.owl`. The browser tabs include "Active ontology", "Entities", "Individuals by class", "DL Query", and "SPARQL Query". The SPARQL query is displayed in a purple header:

```
SPARQL query:
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?subject ?label
WHERE { ?subject a owl:Class .
OPTIONAL { ?subject rdfs:label ?label } .
FILTER regex(?label, "cryopre", "i")
} ORDER BY ?subject
```

The results are shown in a table with two columns: "subject" and "label".

subject	label
cryopreserved human hepatocyte population	"cryopreserved human hepatocyte population"@en

6.4. att. Protege piemērs ar DIDEO ontoloģiju

7. ONTOLOĢIJU NOVĒRTĒŠANA

Maģistra darba ietvaros tika mēģināts replicēt Tiwari un Abraham [60] izveidoto metodoloģiju ontoloģiju novērtēšanai. Tika izvēlētas 10 dažādas brīvi pieejamas ontoloģijas no NCBO bioportāla – vienas no plašākajām biomedicīnas ontoloģiju krātuvēm. Ontoloģijas tika novērtētas, izmantojot 4 dažādus tiešsaistes un bezsaistes rīkus – Protege, TDDOnto, Themis un OOPs!. Balstoties uz [22, 60], tika izveidoti 20 dažādi testi ontoloģiju novērtēšanai.

7.1. Novērtēšana ar Themis

Zāļu mijiedarbības un pierādījumu ontoloģija (DIDEO) bija pirmā, kas tika ielādēta Themis rīkā. Ielāde Themis ir ļoti vienkārša. Ir nepieciešams ielīmēt ontoloģijas URI, to ielādēt un tālāk ontoloģiju var sākt novērtēt. Novērtēšana ir iespējama gan ielādējot testa komplektu no URL, pievienojot RDF failu, kas izmanto ontoloģiju, gan arī vienkārši ielīmējot testa izteiksmi iekšā Themis. Šajā darbā tika izmantota vienkāršā pieeja, ielīmējot testa izteiksmi pa vienai. Ir iespēja izteiksmes ielīmēt arī saraksta veidā, atdalot tās ar semikolu.

DIDEO ontoloģiju nebija iespējams novērtēt Themis rīkā, jo pēc katra palaistā testa tika izmesta kļūda (7.1. attēls). Interesanti ir tas, ka rakstot testa izteiksmes Themis atpazīna ontoloģijā esošās klases (7.2. attēls). Taču, pēc testa palaišanas nekāds rezultāts, izņemot izmesto kļūdu, netika atgriezts. Nav skaidrs, kāpēc testi DIDEO ontoloģijai netika izpildīti un ko šāda kļūda nozīmē, vai ko ar tādu darīt.

Load from file
<http://purl.obolibrary.org/obo/dideo/release/2022-06-14/dideo.owl> See the glossary of terms Remove

2 Check the tests
 You can load a test suite from a URL, adding an RDF file that uses the VTC ontology.

Load from URI

Or you can copy the test suite code, if the test suite uses the VTC ontology.

Load from file

Or you can add the tests directly. To add more than one test separate them by using ";\n".

The following link shows all the supported tests. In this other link you can also find some examples that can be useful to propose tests.

Check

Something went wrong. Check that the ontology doesn't have any inconsistencies or unsatisfiable classes


7.1. att. DIDEO ontoloģijas testa piemērs un Themis kļūda

Check

DIDEO_00000100
 DIDEO_00000101
 DIDEO_00000102
 DIDEO_00000103
 DIDEO_00000104
 DIDEO_00000105
 DIDEO_00000106
 DIDEO_00000107
 DIDEO_00000108
 DIDEO_00000109
 DIDEO_00000110
 DIDEO_00000111
 DIDEO_00000112
 DIDEO_00000113
 DIDEO_00000114
 DIDEO_00000115
 DIDEO_00000116
 DIDEO_00000117
 DIDEO_00000118
 DIDEO_00000119
 DIDEO_00000120
 DIDEO_00000121
 DIDEO_00000122
 DIDEO_00000123
 DIDEO_00000124
 DIDEO_00000125
 DIDEO_00000126
 DIDEO_00000127
 DIDEO_00000128
 DIDEO_00000149

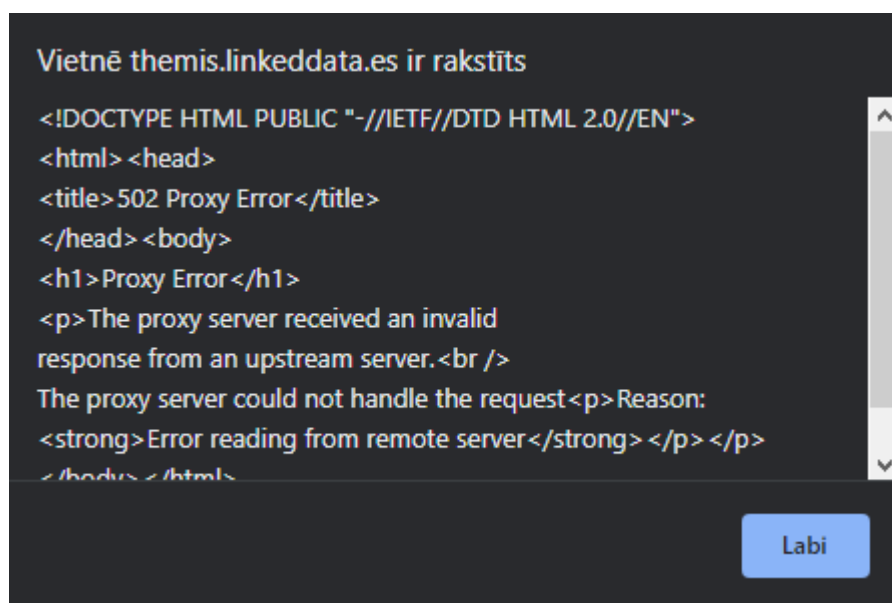
s (e.g., type) are keywords that
 the type (e.g., class or individual)

Clean results
Export test suite



7.2. att. Themis piedāvā pieejamo DIDEO ontoloģijas klašu sarakstu

Biomedicīnas pētījumu ontoloģija (OBI) bija otrā ontoloģija, kas tika novērtēta ar Themis. OBI ielāde Themis arī sagādāja grūtības. Pirmajos mēģinājumos OBI ielāde radīja kļūdu Themis mājaslapā (7.3. attēls). Šo mēģinājumu laikā tika mēģināts palaist OBI ontoloģiju arī no cita datora ar citu interneta pieslēgumu, kas noveda pie mājaslapas īslaicīga nepieejamības (1. pielikums). Nav zināms, vai tas notika tieši OBI ontoloģijas dēļ, vai tomēr Themis rīks tiek ikdienā ilgstoši noslogots un nav spējīgs apstrādāt tik daudz pieprasījumu. Tika mēģināts OBI ielādēt arī vēlāk, kas rezultātā noveda vai nu pie tās pašas mājaslapas kļūdas, vai arī pie DIDEO ontoloģijai līdzīgas kļūdas. Otrajā gadījumā Themis atkal bija spējīgs atpazīt ontoloģijā esošās klases un tās piedāvāt lietotāja izvēlei testu rakstīšanas laikā, taču pēc testa palaišanas tika izmesta jau minētā kļūda.



```
Vietnē themis.linkeddata.es ir rakstīts
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html> <head>
<title>502 Proxy Error</title>
</head> <body>
<h1>Proxy Error</h1>
<p>The proxy server received an invalid
response from an upstream server.<br />
The proxy server could not handle the request<p>Reason:
<strong>Error reading from remote server</strong></p></p>
</body> </html>
```

Labi

7.3. att. Themis kļūda pēc OBI ontoloģijas ielādes

Pierādījumu un secinājumu ontoloģijai (ECO) pirmie 6 testi bija veiksmīgi. Netika atklāta neviena problēma veicot pārbaudi 3 dažādām klasēm un 3 dažādām īpašībām. Taču tālākos testus atkal nebija iespējams veikt, jo kārtējo reizi Themis parādīja kļūdu, ka kaut kas nogāja greizi (1. pielikums).

Genotipa ontoloģijai (GENO) pārbaudot klašu esamību, viena pārbaude no trim bija veiksmīga. GENO_000022 klase veiksmīgi nokārtoja pārbaudi, taču pārējās divas klases Themis uzskatīja par nedefinētām. Līdzīgi ar īpašībām – tikai viena (IAO_0000219) no trim nokārtoja pārbaudi. Pārējās arī tika uzskatītas par nedefinētām. Visas pārējās GENO pārbaudes nebija veiksmīgas. Rezultāts visām pārbaudēm vienāds - nedefinēti termini (1. pielikums).

Veselības uzraudzības ontoloģijai (HSO) visi 3 īpašību testi bija veiksmīgi un bez problēmām. Visiem pārējiem testiem rezultāts bija vienāds, ka termini testā nav definēti.

Līdzīgi kā ECO arī infekcijas slimību ontoloģijai (IDO) pirmie 6 testi bija veiksmīgi un bez kļūdām. Visiem pārējiem testiem bija vienāds rezultāts, ka kaut kas nogāja greizi.

Testu rezultātu ziņā vislielāko dažādību varēja apskatīt pārbaudot garīgo slimību ontoloģiju (MFOMD). No 20 dažādiem testiem, 18 bija veiksmīgi un bez kļūdām. Viena kļūda bija saistīta ar BFO_0000062 pārbaudi, jo šādu terminu Themis uzskatīja par nenodefinētu. Otra kļūda bija saistīta ar divdesmito testu, kurā tika pārbaudīts vai īpašībai ir diapazons. Šādas prasības netika atrastas ontoloģijā.

Audu un enzīmu avotu ontoloģijai (BTO) tika izveidoti un veikti tikai 12 testi, jo nesavienotību, indivīdu, domēnu un diapazonu izmantošana šajā ontoloģijā neeksistē. Visi 12 testi bija veiksmīgi un bez kļūdām.

Klīnisko mērījumu ontoloģijai (CMO) tika izveidoti un veikti tikai 10 testi. Ontoloģijā tika atrasta tikai viena īpašība. Nesavienotību, indivīdu, domēnu un diapazonu izmantošana šajā ontoloģijā neeksistē. Pirmie septiņi testi bija veiksmīgi un bez kļūdām. Attiecību pārbaudēs starp divām klasēm netika atrastas eksistējošas attiecības. Līdz ar to 10., 11. un 12. testa prasības šajā ontoloģijā nav īstenotas.

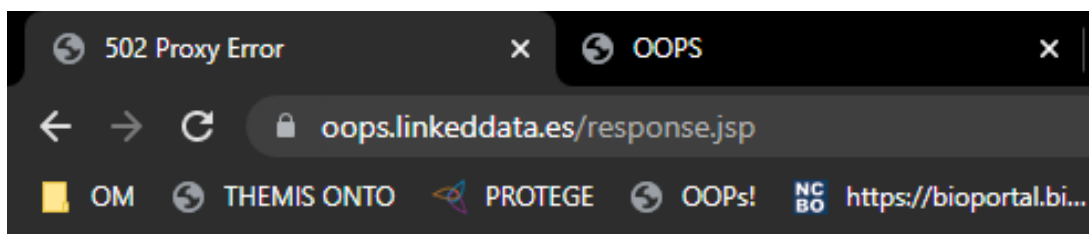
Situācija ar cilvēka slimību ontoloģijas (DOID) pārbaudi Themis rīkā bija tāda pati kā ar OBI ontoloģijas izmantošanu. Pēc ielādes pogas nospiešanas pagāja pāris minūtes un tika parādīta Themis servera kļūda. Līdz ar to testu izpilde nebija iespējama.

Visas minētās pārbaudes ir iespējams detalizēti apskatīt 1. pielikumā.

7.2. Novērtēšana ar OOPs!

Analizējot DIDEO ontoloģiju OOPs! atklāja 14 dažādas problēmas. No tām 7 bija mazsvarīga līmeņa, 4 svarīga līmeņa un 3 kritiskas problēmas. Turklāt, OOPs! parādīja arī ieteikumu saistībā ar simetriskām vai tranzitīvām objekta īpašībām. Šim ieteikumam tika atrasti 3 dažādi gadījumi.

Līdzīgi kā ar Themis rīku arī OOPs! netika galā ar OBI ontoloģiju. Tika parādīta līdzīga starpniekservera kļūda (7.4. attēls). Tāpēc ontoloģija netika pārbaudīta ar OOPs! rīku.



Proxy Error

The proxy server received an invalid response from an upstream server.
The proxy server could not handle the request

Reason: **Error reading from remote server**

Apache Server at oops.linkeddata.es Port 443

7.4. att. OOPS! kļūda pēc OBI ontoloģijas ielādes mēģinājuma

Pārbaudot ECO ontoloģiju ar OOPS! radās tādi pati kļūda kā ar OBI ontoloģijas pārbaudi. Līdz ar to ECO ontoloģijas novērtēšana ar OOPS! nebija iespējama.

Pārbaudot GENO ontoloģiju ar OOPS! tika atklātas 13 dažādas kļūdas. 7 no šīm kļūdām bija mazsvarīgas, 4 bija svarīgas un 2 uzrādījās kā kritiskas.

HSO ontoloģijas pārbaude atklāja 12 dažādas kļūdas un 1 ieteikumu. 6 kļūdas bija mazsvarīgas, 3 kļūdas bija svarīgas un 3 kļūdas bija kritiska līmeņa.

IDO ontoloģijas rezultāts OOPS! rīkā rezultējās ar 10 kļūdām. 5 no tām bija mazsvarīgas, 3 svarīgas un 2 kritiskas.

MFOMD ontoloģijai tika atklātas 14 dažādas kļūdas. 7 no tām bija mazsvarīgas, 5 bija svarīgas un 2 bija kritiskas.

Līdzīgi kā ar OBI un ECO ontoloģijām, BTO bija trešā ontoloģija, kuru nebija iespējams pārbaudīt ar OOPS! rīku.

CMO ontoloģijas rezultāts OOPS! rīkā bija salīdzinoši labs. Kopā tika atklātas trīs kļūdas no kurām divas kritiskas un viena svarīga.

Novērtējot DOID ontoloģiju ar OOPS! tika atklāta tikai viena mazsvarīga kļūda.

Visas minētās pārbaudes ir iespējams detalizēti apskatīt 1. pielikumā.

7.3. Novērtēšana ar TDDOnto

DIDEO ontoloģijas pārbaudes rezultātā TDDOnto pozitīvi apstiprināja visus testus no 7. līdz 13. Visiem pārējiem testiem rezultāts bija vienāds – termini nav nodefinēti.

OBI ontoloģijas TDDOnto pārbaudes rezultāts bija līdzīgs ar DIDEO ontoloģiju. Pārbaudot subsumpciju starp divām klasēm, 2 testi bija veiksmīgi. Pārbaudot attiecības starp divām klasēm visi testi bija veiksmīgi. Pārbaudot nesavienotību starp divām klasēm visi testi bija veiksmīgi. Taču pārējiem testiem rezultāts bija vienāds – termini nav nodefinēti.

ECO ontoloģijas pirmo sešu testu rezultāti bija vienādi – termini nav nodefinēti. Klašu nesavienotības testu rezultāti bija veiksmīgi. Subsumpcijas un klašu attiecību testi arī bija veiksmīgi. Pārējo testu rezultāti atkal parādīja, ka termini nav nodefinēti.

GENO, HSO un IDO ontoloģiju rezultāti bija ļoti līdzīgi. Visām ontoloģijām klašu attiecību testi un subsumpcijas testi bija veiksmīgi. HSO un IDO nesavienotības testi bija veiksmīgi. Taču GENO viens no trim nesavienotības testiem nebija veiksmīgs. Kāds vai abi no BFO_000002 un BFO_000003 terminiem tika atklāti kā nedefinēti.

MFOMD ontoloģijas rezultāti bija mazāk interesanti salīdzinot ar citām ontoloģijām. Visi testi izņemot vienu bija neveiksmīgi. 8. tests bija veiksmīgs, pārējos tika uzrādīts, ka termini nav nodefinēti.

BTO un CMO ontoloģiju rezultāti bija līdzīgi. Divu klašu subsumpcijas testi un klašu attiecību testi bija veiksmīgi un bez kļūdām. Pārējiem testiem vienāds rezultāts – nedefinēti termini.

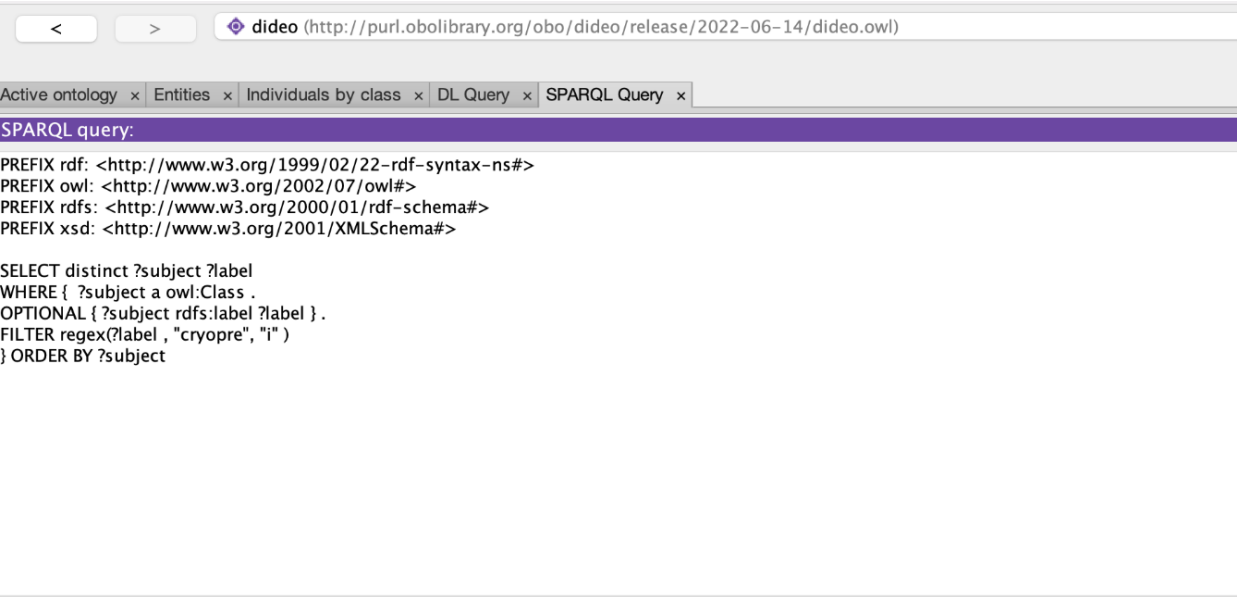
DOID ontoloģiju nebija iespējams pārbaudīt ar TDDOnto, jo palaižot argumentētāju katru reizi rezultāts bija kļūda.

Visas minētās pārbaudes ir iespējams detalizēti apskatīt 1. pielikumā.

7.4. Novērtēšana ar Protege

Novērtēt ontoloģijas ar Protege aizņēma visvairāk laika. Turklāt šis darbs bija visneērtākais, ja salīdzina ar iepriekšējiem ontoloģiju novērtēšanas rīkiem. Tiwari un Abraham [60] minēja divus dažādus veidus, kā novērtēt ontoloģijas ar Protege. Izmantojot argumentētāju, vai izmantojot SPARQL vaicājumus. Šajā maģistra darbā pamatā tika izmantots otrais veids. Taču brīžos, kad SPARQL vaicājums neatrada pārbaudāmos terminus, novērtēšana tika veikta arī ar pirmo metodi.

Ir svarīgi pieminēt, ka starp to brīdi, kad tika vērtētas ontoloģijas ar Protege un brīdi, kad šie rezultāti tika aprakstīti šajā darbā pagāja vairākas dienas. Kaut arī bildes tika izveidotas uzreiz, to aprakstu un analizēšanu autors veica vairākas dienas vēlāk. Kā rezultātā atklājās ļoti interesanta nianse. Pārbaudot DIDEO ontoloģiju pirmo reizi, vairāk par pusi no testiem nebija veiksmīgi. Taču pārbaudot DIDEO dažas dienas vēlāk rezultāti izrādījās pilnīgi pretēji. Piemēram, pirmās testēšanas reizes pirmā testa SPARQL vaicājums neatrada nevienu klasi (7.5. attēls). Taču otrajā novērtēšanas reizē tāds pats SPARQL vaicājums atrada pareizo klasi (7.6. attēls). 13., 14. un 15. testu rezultāti bija veiksmīgi jau pirmajā reizē.



The screenshot shows the Protege SPARQL Query interface. The browser address bar displays the URL: <http://purl.obolibrary.org/obo/dideo/release/2022-06-14/dideo.owl>. The interface includes tabs for "Active ontology", "Entities", "Individuals by class", "DL Query", and "SPARQL Query". The "SPARQL query:" section contains the following query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?subject ?label
WHERE { ?subject a owl:Class .
OPTIONAL { ?subject rdfs:label ?label } .
FILTER regex(?label , "cryopre", "i" )
} ORDER BY ?subject
```

Below the query, the results are displayed in a table with two columns: "subject" and "label".

7.5. att. DIDEO ontoloģijas pirmā testa rezultāts pārbaudot ar Protege

The screenshot shows a web browser window with the URL `http://purl.obolibrary.org/obo/dideo/release/2022-06-14/dideo.owl`. The interface includes tabs for 'Active ontology', 'Entities', 'Individuals by class', 'DL Query', and 'SPARQL Query'. The SPARQL query is as follows:

```

SPARQL query:
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?subject ?label
WHERE { ?subject a owl:Class .
OPTIONAL { ?subject rdfs:label ?label } .
FILTER regex(?label , "cryopre*", "i")
} ORDER BY ?subject

```

The results table shows two columns: 'subject' and 'label'. The first row contains the value 'cryopreserved human hepatocyte population' for both columns.

7.6. att. DIDEO ontoloģijas otrā testa rezultāts pārbaudot ar Protege

Otrajā novērtēšanas reizē netika atklāta neviena problēma un visi testi tika veiksmīgi izpildīti. Tāpēc par DIDEO ontoloģijas Protege novērtēšanas rezultātu tiks uzskatīts otrais testēšanas piegājiens.

OBI, DOID un BTO ontoloģijām neviena no 20 prasībām netika apstiprinātas ar Protege. Taču te ir jāpiemin viens svarīgs novērojums, ka katru reizi, kad šīs ontoloģijas tika ielādētas, Protege strādāja ļoti lēni, bieži uzkāras un mēdza iesalt un nereaģēt.

ECO ontoloģijas novērtēšana ar Protege bija veiksmīga. Visi testa gadījumi tika atrasti bez kļūdām.

GENO ontoloģijai notika līdzīgi kā ar DIDEO. Pirmo reizi veicot novērtēšanu daudzu testu rezultāts bija negatīvs, nepieciešamā informācija netika atrasta. Taču veicot tos pašu testus dažas dienas vēlāk rezultāts uzlabojās. Otrajā novērtēšanas reizē GENO ontoloģijai visi testi bija veiksmīgi.

HSO ontoloģijas gadījumā visas prasības izņemot 15. testu tika apstiprinātas ar Protege.

Novērtējot IDO ontoloģiju netika atrasts rezultāts priekš 9. un 10. prasības. Visas pārējās prasības tika apstiprinātas.

MFOMD ontoloģijas gadījumā netika atrasts rezultāts priekš 19. prasības. Visas pārējās prasības ontoloģijā tika atrastas.

CMO ontoloģijai visas prasības tika atrastas ar Protege.

Visām ontoloģijām novērtēšana notika ar vienādu SPARQL vaicājumu sarakstu. Tāpēc nav skaidrs, kāpēc pirmajā novērtēšanas reizē vairākām ontoloģijām SPARQL vaicājumi neatgriezta nekādu rezultātu.

Visas minētās pārbaudes ir iespējams detalizēti apskatīt 1. pielikumā.

REZULTĀTI

Šajā darbā tika apskatīts apjomīgs literatūras daudzums par ontoloģiju novērtēšanas tehnikām kā arī ontoloģiju novērtēšanas rīkiem. Tālāk tika apskatīti pētījumi saistībā ar NCBO bioportālā atrodamām ontoloģijām to novērtēšanu un kvalitāti. Visbeidzot, tika atkārtots Tiwari un Abraham [60] pētījums, izmantojot daudz reālistiskāku vidi. Tika izmantotas 10 dažādas ontoloģijas no NCBO bioportāla, kurā tās ir brīvi pieejamas. Ontoloģijas tika novērtētas ar 4 dažādiem ontoloģiju novērtēšanas rīkiem. Divi no tiem (Themis un TDDOnto) tika izmantoti ontoloģiju zināšanu verifikācijai. Ar pārējiem diviem (Protege un OOPs!) tika veikta ontoloģiju zināšanu validācija. Katrai no ontoloģijām tika izstrādāti 20 dažādi pārbaudes gadījumi, kas tika pārbaudīti ar trīs rīkiem (Themis, Protege, TDDOnto). Turklāt OOPs! tika izmantots, lai papildus pārliecinātos par tādiem ontoloģijas aspektiem kā pilnīgums, pareizība, lakoniskums, konsekvence un skaidrība.

Vairāki autori [4, 5, 6, 7] līdz šim ir veikuši NCBO bioportālā atrodamo ontoloģiju kvalitātes izpēti. Kaut gan Tiwari un Abraham [60] pētījumā netika izmantotas brīvi pieejamas ontoloģijas, arī šo autoru pētījums atklāja, ka, izņemot autoru pašu veidoto ontoloģiju, neviena no izvēlētajām ontoloģijām neizturēja kvalitātes pārbaudi. Arī šajā maģistra darbā ontoloģiju novērtēšanas rezultāti ir līdzīgi.

Vislielāko pārsteigumu sagādāja DIDEO, OBI un DOID ontoloģijas, kuras ar Themis rīku nebija iespējams pārbaudīt, jo Themis vai nu pārstāja darboties īsi pēc pārbaudes, vai lādējās tik ilgi līdz izmeta kļūdu. ECO un IDO novērtēšana bija savāda. Daži pārbaudes gadījumi bija veiksmīgi, taču, pārbaudot citus, rezultāts bija vienāds, ka kaut kas nogāja greizi. Bija arī labi piemēri, novērtējot ar Themis rīku. Tādas ontoloģijas kā BTO, MFOMD un CMA parādīja diezgan labu rezultātu. BTO ontoloģijai ar Themis neatklājās neviena problēma, MFOMD bija 2 neveiksmīgi pārbaudes gadījumi, bet CMO ontoloģijai tādi bija trīs.

OBI, ECO un BTO ontoloģijas nebija iespējams pārbaudīt arī ar OOPs! rīku. Novērtējot DIDEO ontoloģiju ar OOPs! rīku rezultāts bija 14 dažādas kļūdas, kas, iespējams, ir atbilde tam, kāpēc ar Themis pārbaudīt neizdevās. Vislabāk ar OOPs! veicās DOID un CMO ontoloģijām. Vienai tika atklātas tikai viena kļūda, bet otrai trīs.

Pārbaudot ontoloģijas ar TDDOnto, rezultāti bija mazāk interesanti kā ar Themis vai OOPs!. Visām ontoloģijām, izņemot DOID, kuru nebija iespējams pārbaudīt, daļa pārbaudes gadījumu bija veiksmīgi, bet daļa uzrādīja problēmu, ka termini nav nodefinēti.

Pārbaudot ar Protege rīku, daļai no ontoloģijām nebija neviena veiksmīga pārbaudes gadījuma (OBI, DOID, BTO). Dažām ontoloģijām bija līdzīgi rezultāti kā ar TDDOnto, kur daļa pārbaudes gadījumu bija veiksmīgi un daļa neveiksmīgi. ECO un CMO ontoloģijām visi pārbaudes gadījumi bija veiksmīgi.

No darbā iegūtajiem rezultātiem var secināt, ka tie ir līdzīgi tiem rezultātiem, kurus var atrast citos, ar ontoloģiju novērtēšanu saistītos, pētījumos. Neviena no izvēlētajām ontoloģijām nenokārtoja visus pārbaudes gadījumus. Dažām ontoloģijām pārbaudes gadījumi bija labāki ar vienu rīku, bet sliktāki ar otru.

Šo darbu ir iespējams attīstīt tālāk, pētot gan virzienus saistītus ar šīs metodes metodoloģijas uzlabošanu, gan arī mēģinot attīstīt automatizētu ontoloģiju novērtēšanas pieeju, kas ņemtu vērā visus šajā darbā minētos ontoloģiju novērtēšanas aspektus.

Visbeidzot, šajā darbā uzstādītie mērķi tika sasniegti. Tika apkopots liels daudzums ar dažādu literatūru un atkārtots Tiwari un Abraham [60] pētījums.

SECINĀJUMI

Viens no galvenajiem secinājumiem šim darbam ir tāds, ka katrai novērtēšanas pieejai ir savs mērķis. Mērķi var būt dažādi. Piemēram, Tiwari un Abraham [60] izvēlējās ontoloģiju novērtēt uzreiz pēc izveides un pirms tās publiskošanas. Šī ir viena no lietām par ko ir rakstījuši arī Amith et al. [7] saistībā ar NCBO bioportālā pieejamajām ontoloģijām. Daudzām no portālā pieejamajām ontoloģijām nepastāv dokumentēts pierādījums, ka ontoloģijas ir novērtētas pirms to publiskošanas. Bieži, ontoloģiju novērtēšana nav dokumentēta vispār.

Cits mērķis ir mēģināšana izvēlēties starp vairākām ontoloģijām. Šādos gadījumos noteikti izvēle netiek veikta tikai kvalitātes vērtējuma dēļ. Tiek apskatīti vairāki faktori, kā piemēram, ontoloģijas piemērotība konkrētajam mērķim (darbam vai projektam, vai citam mērķim). Šādos gadījumos nebūtu prātīgi izmantot apskatīto pieeju. Tā ir pārāk lēna un ir nepieciešams daudz resursi, lai kārtīgi apskatītu pat vienu ontoloģiju.

Cits ontoloģiju novērtēšanas mērķis var būt tad, kad ontoloģija ir atrasta un ir vēlme to lietot, taču nav skaidra tās kvalitāte. Varbūt jau ir izmantots kāds no citiem rīkiem, lai izvēlētos starp 5 dažādām līdzīgām ontoloģijām. Izvēle ir veikta, taču nav skaidrs, kādā stāvoklī ir šī ontoloģija un ir nepieciešamība novalidēt un verificēt tās kvalitāti.

Šī pieeja vislabāk derēs tādiem gadījumiem, kad ontoloģija jau ir izvēlēta un to vajag novērtēt. Iespējams, derēs arī gadījumiem, kad kvalitātes vērtējums jāveic tieši pēc ontoloģijas izveides. Pastāv iespēja, ka kādam šī pieeja derēs arī nepārtrauktai kvalitātes nodrošināšanai. Piemēram, ja pārbaudes gadījumi jau eksistē un šī pieeja jau ir izmantota pirms tam, tad ontoloģijas jaunās versijas ir iespējams novērtēt, lai pārlicinātos, ka nekas nav sabojāts, vai ka nav izveidojušās kādas problēmas. Tāpat, noteikti šādā veidā ir iespējams pārbaudīt ontoloģiju pēc tam, kad ontoloģijas zināšanas ir papildinātas ar kādu citu ontoloģiju. Šajā gadījumā atkal novērtēšana tiks veikta kvalitātes nodrošināšanas ietvaros.

Šai pieejai ir vairāki trūkumi. Viens no galvenajiem trūkumiem ir tāds ka tās izmantošanai ir nepieciešami četri dažādi rīki. Šis ir liels ierobežojums vairāku iemeslu dēļ. Pirmkārt, tas ir ļoti laikietilpīgs process. Visi šie rīki ir jāiestata un ar visiem šiem rīkiem ir jāiemācās strādāt. Arī pats novērtēšanas process ir ļoti laikietilpīgs, jo ir jāraksta vairāki desmiti pārbaudes gadījumu un visi vēlāk jāpārbauda uz vairākiem rīkiem. Pastāv arī iespēja, ka viena kļūda tiek atklāta vienā rīkā, taču netiek atklāta otrā. Piemēram, balstoties uz Tiwari un Abraham [60], Themis un TDDOnto abi pārbauda verifikāciju, balstoties uz vienu un to pašu testu gadījumu sarakstu. Šādos gadījumos

nav skaidrs, kuram rīkam uzticēties vairāk. Otrkārt, kā autors ir pats pārliecinājies, visi rīki nepārtraukti tiek uzlaboti, tiem parādās jaunas versijas. Spraudņu gadījumā ir nepieciešams no jauna izveidot vidi, lai varētu strādāt. Diemžēl, ne visos gadījumos šīs jaunās versijas ir detalizēti aprakstītas, un ne vienmēr ir skaidrs, kā pāriet uz jaunāku versiju. Piemēram, TDDOnto gadījumā, jaunā versija nav pieejama tiešsaistē, vai arī to nevar viegli atrast. Līdz ar to, autors ir izmantojis brīvi pieejamo, taču veco versiju.

Otrs ierobežojums ir tāds, ka ne visi no apskatītajiem rīkiem ir spējīgi strādāt ar ļoti milzīgām ontoloģijām. Themis gadījumā, mājaslapa vienkārši nestrādā. Nav arī skaidrs, kāpēc tā nestrādā konkrētos gadījumos. Vai tā ir konkrētās ontoloģijas problēma, vai tomēr pats rīks nav spējīgs apstrādāt tik apjomīgus datus. TDDOnto un OOPs! gadījumā, var gadīties gaidīt ļoti ilgi, lai dabūtu atbildi. Arī Protege rīkam nevienmēr varēja uzticēties. Ar dažām ontoloģijām Protege novērtēšana bija ļoti lēna un rīks bieži iesaldējās (piemēram, OBI, DOID un BTO gadījumos).

Treškārt, līdz galam nav skaidrs vai šāda pieeja derēs milzīgu ontoloģiju novērtēšanai. Autori [60] savā darbā koncentrējās uz SHCO ontoloģiju kurai, balstoties uz [54], ir mazāk par 100 klasēm. Pieņemot, ka šāda apjoma ontoloģijai pietiek ar 20 dažādiem testa gadījumiem, nav skaidrs, cik lielu testa gadījumu sarakstu būtu jāveido priekš ontoloģijām, kuru klašu skaits ir mērāms vairākos tūkstošos. Diemžēl, šobrīd šī pieeja ir salīdzinoši jauna, un pietrūkst pētījumi, kas koncentrētos tieši uz šādas pieejas testa gadījumu rakstīšanu.

Visbeidzot, ir jautājumi par testa gadījumu veidošanas procesu. Kaut arī autori [60] savos testa gadījumos ir iekļāvuši dažādas pārbaudes tēmētas dažādiem mērķiem, nav skaidrs, uz ko autori balstījās, lai nonāktu pie izvēlētajiem pārbaudes mērķiem. Piemēram, Themis [59] mājaslapā pieejamais atbalstīto testu saraksts satur 27 dažādus testus. Taču, autoru [60] darbā izveidotais saraksts satur 20 testus, no kuriem daudzi atkārtojās. Šeit eksistē arī cits izaicinājums – kā pareizi izveidot testu sarakstu ontoloģijai, ar kuru nekad nav strādāts un kuras saturs nav zināms? Vai ir pareizi mēģināt iekļaut visus rīkos atbalstītus testus, vai tomēr koncentrēties uz kādiem ontoloģijai specifiskiem aspektiem?

Šajā darbā aprakstīto metodi ir iespējams pētīt un attīstīt tālāk. Daudzi pētījumi ontoloģiju izveidošanas un novērtēšanas jomā ir rakstījuši par automatizētu novērtēšanas rīku trūkumu. Šo metodi varētu attīstīt tālāk un no tās izveidot vienu rīku, kas apvieno šajā darbā apskatītos validēšanas un verificēšanas mehānismus. Šāds rīks palīdzētu daudz ātrāk un efektīvāk realizēt ontoloģiju novērtēšanas procesu.

Izstrādājot šo maģistra darbu, autors apguva prasmes strādāt ar tādiem ontoloģiju novērtēšanas rīkiem kā Themis, OOPs!, Protege un TDDOnto. Turklāt, autors iemācījās rakstīt SPARQL vaicājumus Protege rīkā. Līdz šim autors nekad nebija strādājis ar Themis, OOPs! un TDDOnto. Turklāt, autoram bija ļoti limitētas zināšanas par Protege rīku un SPARQL vaicājumu valodu.

IZMANTOTĀ LITERATŪRA UN AVOTI

- [1] E. Alasaarela, R. Nemama, S. Demello, “Drivers and Challenges of Wireless Solutions in Future Healthcare,” *Int. Conf. On eHealth, Telemed. and Soc. Med.* (eTELEMED 2009), 2009, pp. 19-24. Pieejams:
https://www.researchgate.net/publication/221222803_Drivers_and_Challenges_of_Wireless_Solutions_in_Future_Healthcare (aplūkots: 09.01.2022)
- [2] A. Algergawy, B. Konig-Ries, “Partitioning of BioPortal Ontologies: An Empirical Study,” in *12th Int. Conf. On Semantic Web App. and Tools in Healthcare and Life Sciences (SWAT4HCLS 2019)*, 2019. Pieejams: <http://ceur-ws.org/Vol-2849/paper-10.pdf>
- [3] S. Allen, “2022 Global Health Care Outlook,” *Deloitte*, 2022. Pieejams:
<https://www2.deloitte.com/global/en/pages/life-sciences-and-healthcare/articles/global-health-care-sector-outlook.html> (aplūkots: 10.01.2022.).
- [4] M. Amith, C. Tao, “A Web Application Towards Semiotic-based Evaluation of Biomedical Ontologies,” in *The 14th Int. Semantic Web Conf.*, 2015. Pieejams:
https://ceur-ws.org/Vol-1428/BDM2I_2015_paper_5.pdf
- [5] M.T. Amith, C. Tao, “Modulated Evaluation Metrics for Drug-Based Ontologies,” *Journal of Biomedical Semantics*, Springer, vol. 8, no. 1, 2017. Pieejams:
https://www.researchgate.net/publication/315297911_Modulated_Evaluation_Metrics_For_Drug-Based_Ontologies
- [6] M. Amith, F. Manion, C. Liang, M. Harris, D. Wang, Y. He, C. Tao, “Architecture and Usability of OntoKeeper, an Ontology Evaluation Tool,” *BMC Medical Informatics and Decision Making*, vol. 19, no. 4, Springer, 2019. Pieejams:
<https://bmcmmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-019-0859-z>
- [7] M. Amith, Z. He, J. Bian, J.A. Lossio-Ventura, C. Tao, “Assessing the Practice of Biomedical Ontology Evaluation: Gaps and Opportunities,” *Journal of Biomedical Informatics*, vol. 80, 2018, pp. 1-13. Pieejams:
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5882531/>

- [8] ARISTOTELES, *Welcome page*. Pieejams:
<https://semantic.cs.put.poznan.pl/wiki/aristoteles/doku.php> (aplūkots: 12.10.2022.).
- [9] J.C. Arpirez, O. Corcho, M. Fernandez-Lopez, A. Gomez-Perez, “WebODE in a Nutshell,” *Ai Magazine*, vol.24, no.3, AAAI, 2003, pp.37-48. Pieejams:
<https://github.com/oeg-upm/webODE>
- [10] J. Bandeira, I.I. Bittencourt, P. Espinheira, S. Isotani, “FOCA: A Methodology for Ontology Evaluation,” *arXiv*, 2016. Pieejams:
https://www.researchgate.net/publication/311586021_FOCA_A_Methodology_for_Ontology_Evaluation
- [11] C. Bock, A. Fokoue, P. Haase, R. Hoekstra, I. Horrocks, A. Ruttenberg, U. Sattler, M. Smith, *OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (Second Edition)*, W3C Recommendation, 2012. Pieejams:
<https://www.w3.org/TR/owl2-syntax/>
- [12] J. Brank, M. Grobelnik, D. Mladenic, “A survey of ontology evaluation techniques,” *Proceedings of the Conference on Data Mining and Data Warehouses*, 2005. Pieejams:
https://ai.ia.agh.edu.pl/wiki/_media/pl:miw:2009:brankevaluationsikdd2005.pdf
- [13] J. Brank, M. Grobelnik, D. Mladenic, “A survey of ontology evaluation techniques,” 2009. Pieejams:
<https://ailab.ijs.si/dunja/SiKDD2005/Papers/BrankEvaluationSiKDD2005.pdf>
- [14] C. Brewster, H. Alani, S. Dasmahapatra, Y. Wilks, “Data Driven Ontology Evaluation,” 2004. Pieejams: <http://www.lrec-conf.org/proceedings/lrec2004/pdf/737.pdf>
- [15] A. Burton-Jones, V.C. Storey, V. Sugumaran, P. Ahluwalia, “A Semiotic Metrics Suite for Assessing the Quality of Ontologies,” *Data & Knowledge Engineering*, vol.55, no.1, Elsevier, 2005, pp.84-102. Pieejams:
<https://www.sciencedirect.com/science/article/abs/pii/S0169023X0400223X>
- [16] Centers for Medicare & Medicaid Services, *Historical*. Pieejams:
<https://www.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Trends-and-Reports/NationalHealthExpendData/NationalHealthAccountsHistorical> (aplūkots: 12.01.2022.).

- [17] Data & Knowledge Group, *HermiT OWL Reasoner*. Pieejams: <http://www.hermit-reasoner.com/> (aplūkots: 14.09.2022.).
- [18] B.A. Dawood, M. Sah, “Semantic Web and Healthcare System in IoT Enabled Smart Cities” *Innovations in Smart Cities Applications*, vol. 4, Springer, 2021, pp.546-557. Pieejams: https://www.researchgate.net/publication/349248942_Semantic_Web_and_Healthcare_System_in_IoT_Enabled_Smart_Cities
- [19] R.Q. Dividino, M.Romanelli, D.Sonntag, “Semiotic-based Ontology Evaluation Tool (S-OntoEval),” in *Proc. of the Int. Conf. On Language Resources and Evaluation (LREC 2008)*, 2008. Pieejams: https://www.researchgate.net/publication/220745980_Semiotic-based_Ontology_Evaluation_Tool_S-OntoEval
- [20] A. Duque-Ramos, J.T. Fernandez-Breis, R.D. Stevens, N. Aussenac-Gilles, “OQuaRE: A SQuaRE-based Approach for Evaluating the Quality of Ontologies,” *Journal of Research and Practice in Information Tenchnology*, vol.43,no.2, Australian Computer Society, 2011, pp.159-176. Pieejams:https://www.researchgate.net/publication/285665382_OQuaRE_A_SQuaRE-based_approach_for_evaluating_the_quality_of_ontologies
- [21] A. Fernandez-Izquierdo, R. Garcia-Castro, “Requirements Behaviour Analysis for Ontology Testing,” in *European Knowledge Acquisition Workshop (EKAW 2018)*, LNAI 11313, Springer, 2018, pp.114-130. Pieejams: https://vicinity2020.eu/vicinity/system/files/publications/requirements_behaviour_analysiss_for_ontology_testing_upm.pdf
- [22] A. Fernandez-Izquierdo, R. Garcia-Castro, “Themis: a Tool for Validating Ontologies Through Requirements,” in *The 31st International Conference on Software Engineering and Knowledge*, 2019, pp. 573-578. Pieejams: https://www.researchgate.net/publication/335152619_Themis_a_tool_for_validating_ontologies_through_requirements
- [23] J. Frey, D. Streitmatter, F. Gotz, S. Hellmann, N. Arndt, “DBpedia Archivo: A Web-Scale Interface for Ontology Archiving Under Consumer-Oriented Aspects,” *Semantic Systems*.

The Power of AI and Knowledge Graphs, LNCS, Springer, 2020. Pieejams:
<https://archivo.dbpedia.org/about>

- [24] A. Gomez-Perez, “Ontology Evaluation,” *Handbook on Ontologies*, International Handbooks on Information Systems, Springer, 2004, pp.251-272.
https://oa.upm.es/6427/1/Ontology_Evaluation_1stEd.pdf
- [25] T.R. Gruber, “A translation approach to portable ontology specifications,” *Knowledge Acquisition*, vol. 5, no. 2, 1993, pp. 199-220. <https://tomgruber.org/writing/ontolingua-kaj-1993.pdf>
- [26] T.R. Gruber, “Ontology,” in *Encyclopedia of Database Systems*, L. Liu, M. Tamer-Oszu, Springer-Verlag, 2009. Pieejams: <https://tomgruber.org/writing/definition-of-ontology.pdf>
- [27] N. Guarino, C.A. Welty, *Handbook on Ontologies*. Berlin, Germany, Springer, 2004. Pieejams:
https://www.researchgate.net/publication/226934944_An_Overview_of_OntoClean
- [28] N. Guarino, D. Oberle, S. Staab, “What Is an Ontology?” *Handbook on Ontologies*, Springer, 2009, pp.1-17. Pieejams:
https://www.researchgate.net/publication/226279556_What_Is_an_Ontology
- [29] M. Halper, C. Ochs, Y. Perl, S. Arabandi, M.A. Musen, “Quality Assurance of Ontology Content Reuse,” in *Proc. of the 9th Int. Conf. On Biological Ontology (ICBO 2018)*, 2018. Pieejams: http://ceur-ws.org/Vol-2285/ICBO_2018_paper_15.pdf
- [30] M. Horridge, B. Parsia, N.F. Noy, M.A. Musen, “Reasoning based quality assurance of medical ontologies: a case study,” *Annual Symposium proceedings*, AMIA Symposium, 2014, pp. 671-680. Pieejams: <https://pubmed.ncbi.nlm.nih.gov/25954373/>
- [31] Z. Huang, F. Van Harmelen, A. Ten Teije, P. Groot, C. Visser, “Reasoning with Inconsistent Ontologies: Framework and Prototype,” *EU-IST Integrated Project*, 2004. Pieejams:
https://www.researchgate.net/publication/240209181_Reasoning_with_Inconsistent_Ontologies_Framework_and_Prototype

- [32] R. Jackson, N. Matentzoglou, J.A. Overton, R. Vita, J.P. Balhoff, P.L. Buttigieg, S. Carbon, M. Courtot, A.D. Diehl, D.M. Dooley, W.D. Duncan, N.L. Harris, M.A. Haendel, S.E. Lewis, D. A. Natale, D. Osumi-Sutherland, A. Ruttenberg, L.M. Schriml, B. Smith, C.J. Stoeckert, N.A. Vasilevsky, R.L. Walls, J. Zheng, C.J. Mungall, B. Peters, “*OBO Foundry in 2021: operationalizing open data principles to evaluate ontologies*,” Database: the journal of biological databases and curation, 2021. Pieejams: <https://obofoundry.org/>
- [33] W. Jin, D. H. Kim, “Design and Implementation of e-Health System Based on Semantic Sensor Network Using IETF YANG,” *Sensors*, vol.18, no.2, 2018, pp.629-648 Pieejams: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5854965/> (aplūkots: 14.01.2022)
- [34] Keet blog, *Improved! TDDonto v2—more types of axioms supported and better feedback*. Pieejams: <https://keet.wordpress.com/2016/12/15/improved-tddonto-v2-more-types-of-axioms-supported-and-better-feedback/> (aplūkots: 05.09.2022.)
- [35] Keet blog, *The TDDonto tool to try out TDD for ontology authoring*. Pieejams: <https://keet.wordpress.com/2016/04/17/the-tddonto-tool-to-try-out-tdd-for-ontology-authoring/> (aplūkots: 18.09.2022.)
- [36] B. Lantow, “OntoMetrics: Application of On-line Ontology Metric Calculation,” *BIR Workshops*, 2016. Pieejams: <https://ceur-ws.org/Vol-1684/paper19.pdf>
- [37] A. Lawrynowicz, C.M. Keet, “The TDDonto Tool for Test- Driven Development of DL Knowledge Bases,” in *Proc. of 29th Int. Workshop on Description Logics (DL 2016)*, 2016. Pieejams: http://ceur-ws.org/Vol-1577/paper_15.pdf
- [38] A. Lozano-Tello, A. Gomez-Perez, “ONTOMETRIC: A Method to Choose the Appropriate Ontology,” *Journal of Database Management*, vol.15, no.2, IGI Global, 2004, pp.1-18. Pieejams: https://www.researchgate.net/publication/220373744_ONTOMETRIC_a_method_to_choose_the_appropriate_ontology
- [39] A. Lozano-Tello, A.Gomez-Perez, “ONTOMETRIC: A Method to Choose the Appropriate Ontology,” *Journal of Database Management*, vol.15,no.2, IGI Global, 2004, pp.1-18. Pieejams: <https://www.igi-global.com/gateway/article/3308>

- [40] A. Maedche, S. Staab, “ Measuring Similarity between Ontologies” in *Proc. of the 13th Int. Conf. on Knowledge Engineering and Knowledge Mgmt.* , 2002, pp. 15-21. Pieejams: https://www.researchgate.net/publication/226450264_Measuring_Similarity_between_Ontologies
- [41] Z. Mahlaza, C.M. Keet, “OntoClean in OWL with a DL reasoner – A tutorial,” 2019. Pieejams: <https://people.cs.uct.ac.za/~mkeet/OEbook/ontocleantutorialOE19.pdf>
- [42] J. Moreira, L.F. Pires, M. Sinderen, L. Daniele, “SAREF4health: IoT Standard-Based Ontology-Driven Healthcare Systems,” *Formal Ontology in Information Systems (FOIS)* ,2018. Pieejams: https://www.researchgate.net/publication/328744088_SAREF4health_IoT_Standard-Based_Ontology-Driven_Healthcare_Systems
- [43] M.A. Musen, “The Protege Project: A Look Back and A Look Forward,” *AI Matters*, vol.1, no.4, 2015, pp.4-12. Pieejams: <https://protege.stanford.edu/>
- [44] A. Nikiforova, S. Tiwari, V. Rovite, N. Kante, J. Klovins, “Evaluation and Visualization of Healthcare Semantic Models,” *Semantic Models in IoT and eHealth Applications*, Ed. S. Tiwari, F.O. Rodriguez, M.A. Jabbar, Academic Press, 2022.
- [45] N.F. Noy, N.H. Shah, P.L. Whetzel, B. Dai, M. Dorf, N. Griffith, C. Jonquet, D.L. Rubin, M.A. Storey, C.G. Chute, M.A. Musen, “BioPortal: Ontologies and Integrated Data Resources at the Click of A Mouse,” *Nucleic Acids Research (Web Server issue)*, vol.37, 2009. Pieejams: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2703982/pdf/gkp440.pdf>
- [46] L. Obrst, W. Ceusters, I. Mani, S. Ray, “The Evaluation of Ontologies,” *Semantic Web*, 2007, pp.139-158. Pieejams: https://www.researchgate.net/publication/225855591_The_Evaluation_of_Ontologies
- [47] C. Ochs, Y. Perl, J. Geller, S. Arabandi, T. Tudorache, M.A.Musen, “An Empirical Analysis of Ontology Reuse in BioPortal,” *Journal of Biomedical Informatics*, vol.72, Elsevier, 2017, pp.165-177. Pieejams: <https://www.sciencedirect.com/science/article/pii/S153204641730120X>

- [48] M. Poveda-Villalon, A. Gomez-Perez, M.C. Suarez-Figueroa, “OOPS! (OntOlogy Pitfall Scanner!): An On-line Tool for Ontology Evaluation,” *International Journal on Semantic Web and Information Systems*, vol. 10, no. 2, 2014, pp. 7-34. Pieejams:
https://www.researchgate.net/publication/285623865_OOPS_OntOlogy_Pitfall_Scanner_An_on-line_tool_for_ontology_evaluation
- [49] M. Poveda-Villalon, A. Gomez-Perez, M.C. Suarez-Figueroa, “OOPS!(Ontology Pitfall Scanner!): An On-line Tool for Ontology Evaluation,” *International Journal on Semantic Web and Information Systems* (IJSWIS), IGI Publishing, 2014, pp.7-37. Pieejams:
<https://oops.linkeddata.es/catalogue.jsp>
- [50] M. Sabou, M. Fernandez, “Ontology (Network) Evaluation,” 2012. Pieejams:
https://www.researchgate.net/publication/288617622_Ontology_Network_Evaluation
- [51] Z. Say, S. Fathalla, S. Vahdati, J. Lehmann, S. Auer, “Ontology Design for Pharmaceutical Research Outcomes,” 2020. DOI: 10.13140/RG.2.2.20218.72644.
- [52] M. Salvadores, R.R. Alexander, M.A. Musen, N.F. Noy, “BioPortal as a Dataset of Linked Biomedical Ontologies and Terminologies in RDF,” *Semant Web*, vol.4, no. 3, 2013, pp.277-284. Pieejams: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4159173/>
- [53] D. Schober, I. Tudose, V. Svatek, M. Boeker, “OntoCheck: verifying ontology naming conventions and metadata completeness in Protégé 4,” *J. Biomed. Semant.*, vol. 3, no. 2, 2012. Pieejams:
https://www.researchgate.net/publication/232224662_OntoCheck_Verifying_ontology_naming_conventions_and_metadata_completeness_in_Protege_4
- [54] Smart Health Care Ontology for Health Care, *Revision: 1.1*. Pieejams:
<https://tiwari2019.github.io/myrepo/OnToology/Healthcare.owl/documentation/index-en.html> (aplūkots: 02.01.2023.).
- [55] S. Sotiriadis, C. Thermolia, E. Bei, K. Stravoskoufos, E. Petrakis, “Designing A Patient Monitoring System Using Cloud and Semantic Web Technologies,” in *17th Int. Conf. On Brain and Health Informatics*, 2015. Pieejams:
https://www.researchgate.net/publication/325665421_Designing_a_patient_monitoring_system_using_Cloud_and_Semantic_web_technologies

- [56] R. Stamper, K. Liu, M. Hafkamp, Y. Ades, “Understanding the roles of signs and norms in organizations - A semiotic approach to information systems design,” *J. Of Behaviour and Inform. Tech.*, vol. 19, no. 1, 2000, pp. 15-27. Pieejams: https://www.researchgate.net/publication/228597699_Understanding_the_roles_of_signs_and_norms_in_organizations_-_A_semiotic_approach_to_information_systems_design
- [57] H. Taleb, A. Nasser, G. Andrieux, N. Charara, E. M. Cruz, “Wireless technologies, medical applications and future challenges in WBAN: a survey,” *Wireless Networks*, vol.27, 2021, pp. 5271–5295. Pieejams: <https://link.springer.com/article/10.1007/s11276-021-02780-2#ref-CR2>
- [58] S. Tartir, I.B. Arpinaru, A. Sheth, “ Ontological Evaluation and Validation,” *Theory and Applications of Ontology:Computer Applications* ,2010, pp. 115-130. Pieejams:https://www.researchgate.net/publication/226387227_Ontological_Evaluation_and_Validation
- [59] Themis, *Description of the supported tests*. Pieejams: <https://themis.linkeddata.es/tests-info.html> (aplükots: 02.01.2023.).
- [60] S. Tiwari, A. Abraham, “Semantic assessment of smart healthcare ontology,” *Int. J. of Web Inf. Sys.*, vol. 16, no. 4, 2020, pp. 475-491.
- [61] S. Tiwari, S. Jain, “Ontologies as A Semantic Model in IoT,” *Int. J. of Comp. and App.*, vol. 42, no. 1, 2018, pp. 1-11. Pieejams: https://www.researchgate.net/publication/326870785_Ontologies_as_a_semantic_model_in_IoT
- [62] D. Vrandecic, “Ontology Evaluation,” Doctoral Thesis, Faculty of Economics, Karlsruhe Institute for Technology, 2010. Pieejams: <https://www.aifb.kit.edu/images/b/b5/OntologyEvaluation.pdf>
- [63] X. Wang, C. Williams, Z.H. Liu, J. Croghan, “Big data management challenges in health research—a literature review,” *Briefings in Bioinformatics*, vol. 20, no. 1, 2019, pp. 156–167. Pieejams: <https://academic.oup.com/bib/article/20/1/156/4077088>
- [64] P.L. Whetzel, N.F. Noy, N.H. Shah, P.R. Alexander, C. Nyulas, T. Tudorache, M.A. Musen, “BioPortal: enhanced functionality via new Web services from the National

Center for Biomedical Ontology to access and use ontologies in software applications,”
Nucleic Acids Res., vol. 39., 2011, pp. 541-545. Pieejams:
<https://bioportal.bioontology.org/ontologies>

- [65] World Health Organization, “Ageing and Health,” 2022. Pieejams:
<https://www.who.int/news-room/fact-sheets/detail/ageing-and-health#:~:text=By%202050%2C%20the%20world's%20population,2050%20to%20reach%20426%20million> (aplūkots: 10.01.2022.).

PIELIKUMI

1. pielikums

Themis rīka mājaslapa pēc OBI ontoloģijas ielādes mēģinājuma.



Service Unavailable

The server is temporarily unable to service your request due to maintenance downtime or capacity problems. Please try again later.

ECO pirmie 6 testi ar Themis un pirmā kļūda.

<http://purl.obolibrary.org/obo/eco.owl> [See the glossary of terms](#) [Remove](#)

2 **Check the tests**
You can load a test suite from a URL, adding an RDF file that uses the VTC ontology.

[Load from URI](#)

Or you can copy the test suite code, if the test suite uses the VTC ontology.

[Load from file](#)

Or you can add the tests directly. To add more than one test separate them by using ",".

The [following link](#) shows all the supported tests. In this [other link](#) you can also find some examples that can be useful to propose tests.

[Check](#)

Something went wrong. Check that the ontology doesn't have any inconsistencies or unsatisfiable classes

Important information

- The test syntax is **case sensitive** (e.g., Sensor and sensor are detected as different terms)
- For generating the tests you have to use the [glossary of terms](#) (you can change it)
- The test **syntax** is available [here](#). You should insert the terms between brackets ("e.g., [Class] → Sensor"), while the **italised terms** (e.g., *type*) are keywords that cannot be changed
- Terms in **red** indicate that they don't exist in the ontology, while the terms in **orange** indicate that the term exist in the ontology but the type (e.g., class or individual) is not the specified in the term
- The **Absent relation** result does not indicate that the test is not passed, but that something may be missing

Tests results

Test	Result	Problem	
RO_0002222 type Property	Passed	None	
ECO_9000000 type Property	Passed	None	
ECO_9000001 type Property	Passed	None	
ECO_0001809 type Class	Passed	None	
ECO_0007742 type Class	Passed	None	
ECO_0007234 type Class	Passed	None	

ECO pirmie 6 testi ar Themis un otrā kļūda.

✓ <http://purl.obolibrary.org/obo/eco.owl>

▼ See the glossary of terms

🗑 Remove

2 Check the tests

You can load a test suite from a URL, adding an RDF file that uses the VTC ontology.

Load from URI

Or you can copy the test suite code, if the test suite uses the VTC ontology.

Load from file

Or you can add the tests directly. To add more than one test separate them by using “;”.

The following link shows all the supported tests. In this other link you can also find some examples that can be useful to propose tests.

Check

Something went wrong. Check that the ontology doesn't have any inconsistencies or unsatisfiable classes

Important information

- The test syntax is case sensitive (e.g., Sensor and sensor are detected as different terms)
- For generating the tests you have to use the glossary of terms (you can change it)
- The test syntax is available here. You should insert the terms between brackets (“e.g., [Class] → Sensor”), while the italicised terms (e.g., *type*) are keywords that cannot be changed
- Terms in red indicate that they don't exist in the ontology, while the terms in orange indicate that the term exist in the ontology but the type (e.g., class or individual) is not the specified in the term
- The **Absent relation** result does not indicate that the test is not passed, but that something may be missing

Tests results

Test	Result	Problem	
RO_0002222 type Property	Passed	None	🗑
ECO_9000000 type Property	Passed	None	🗑
ECO_9000001 type Property	Passed	None	🗑
ECO_0001809 type Class	Passed	None	🗑
ECO_0007742 type Class	Passed	None	🗑
ECO_0007234 type Class	Passed	None	🗑

ECO pirmie 6 testi ar Themis un trešā kļūda.

<http://purl.obolibrary.org/obo/leco.owl> [See the glossary of terms](#) [Remove](#)

2 **Check the tests**
You can load a test suite from a URL, adding an RDF file that uses the **VTC** ontology.

[Load from URI](#)

Or you can copy the test suite code, if the test suite uses the **VTC** ontology.

[Load from file](#)

Or you can add the tests directly. To add more than one test separate them by using ",".

The [following link](#) shows all the supported tests. In [this other link](#) you can also find some examples that can be useful to propose tests.

[Check](#)

Something went wrong. Check that the ontology doesn't have any inconsistencies or unsatisfiable classes

Important information

- The test syntax is **case sensitive** (e.g., **S**ensor and **s**ensor are detected as different terms)
- For generating the tests you have to use the **glossary of terms** (you can change it)
- The **test syntax** is available [here](#). You should insert the terms between brackets ("e.g., [Class] → Sensor"), while the **italised terms** (e.g., *type*) are keywords that cannot be changed
- Terms in **red** indicate that they don't exist in the ontology, while the terms in **orange** indicate that the term exist in the ontology but the type (e.g., class or individual) is not the specified in the term
- The **Absent relation** result does not indicate that the test is not passed, but that something may be missing

Tests results

Test	Result	Problem	
RO_0002222 type Property	Passed	None	
ECO_9000000 type Property	Passed	None	
ECO_9000001 type Property	Passed	None	
ECO_0001809 type Class	Passed	None	
ECO_0007742 type Class	Passed	None	
ECO_0007234 type Class	Passed	None	

Tests results

Test	Result	Problem	
GENO_0000722 subClassOf GENO_0000783 some GENO_0000782	Undefined terms	The terms in the test are not defined in the ontology	
GENO_0000823 subClassOf IAO_0000219 some GENO_0000516	Undefined terms	The terms in the test are not defined in the ontology	
GENO_0000002 subClassOf GENO_0000583 some GENO_0000036	Undefined terms	The terms in the test are not defined in the ontology	
IAO_0000424 type Property	Undefined terms	The terms in the test are not defined in the ontology	
IAO_0000219 type Property	Passed	None	
GENO_0000823 subClassOf IAO_0000219 some GENO_0000516	Undefined terms	The terms in the test are not defined in the ontology	
GENO_0000611 subClassOf GENO_0000899	Undefined terms	The terms in the test are not defined in the ontology	
GENO_0000823 subClassOf GENO_0000897	Undefined terms	The terms in the test are not defined in the ontology	
GENO_0000885 subClassOf GENO_0000823	Undefined terms	The terms in the test are not defined in the ontology	
GENO_0000022 type Class	Passed	None	
GENO_0000702 type Class	Undefined terms	The terms in the test are not defined in the ontology	
GENO_0000921 type Class	Undefined terms	The terms in the test are not defined in the ontology	

Tests results

Test	Result	Problem	
GENO_0000815 disjointWith GENO_0000702	Undefined terms	The terms in the test are not defined in the ontology	
BFO_0000019 disjointWith BFO_0000017	Undefined terms	The terms in the test are not defined in the ontology	
BFO_000002 disjointWith BFO_000003	Undefined terms	The terms in the test are not defined in the ontology	

Clean results

Export test suite

Tests results

Test	Result	Problem
IAO_0000425 type Property	Undefined terms	The terms in the test are not defined in the ontology

Add the code of the ontology to be validated:

✔ <http://purl.obolibrary.org/obo/geno.owl>
▼ See the glossary of terms
🗑 Remove

2 **Check the tests**

You can load a test suite from a URL, adding an RDF file that uses the VTC ontology.

Or you can copy the test suite code, if the test suite uses the VTC ontology.

Or you can add the tests directly. To add more than one test separate them by using ";".

The following link shows all the supported tests. In this other link you can also find some examples that can be useful to propose tests.

GENO_0000484 type IAO_0000102

Something went wrong. Check that the ontology doesn't have any inconsistencies or unsatisfiable classes

- Important information**
- The test syntax is **case sensitive** (e.g., Sensor and sensor are detected as different terms)
 - For generating the tests you have to use the **glossary of terms** (you can change it)
 - The **test syntax** is available [here](#). You should insert the terms between brackets ("e.g., [Class] → Sensor"), while the **italised terms** (e.g., *type*) are keywords that cannot be changed
 - Terms in **red** indicate that they don't exist in the ontology, while the terms in **orange** indicate that the term exist in the ontology but the type (e.g., class or individual) is not the specified in the term
 - The **Absent relation** result does not indicate that the test is not passed, but that something may be missing

Tests results

Test	Result	Problem
GENO_0000966 domain GENO_0000815	Undefined terms	The terms in the test are not defined in the ontology
GENO_0000239 range GENO_0000702	Undefined terms	The terms in the test are not defined in the ontology
GENO_0000921 type Class	Undefined terms	The terms in the test are not defined in the ontology

1 **Load the ontology**

Add the URI of the ontology to be validated:

Add the code of the ontology to be validated:

<http://purl.obolibrary.org/obo/dideo/release/2022-06-14/dideo.owl>

2 **Check the tests**

You can load a test suite from a URL, adding an RDF file that uses the VTC ontology.

Or you can copy the test suite code, if the test suite uses the VTC ontology.

Or you can add the tests directly. To add more than one test separate them by using ";".

The following link shows all the supported tests. In this other link you can also find some examples that can be useful to propose tests.

DIDEO_00000149 type Class

Something went wrong. Check that the ontology doesn't have any inconsistencies or unsatisfiable classes

Add the code of the ontology to be validated:

<http://purl.obolibrary.org/obo/dideo/release/2022-06-14/dideo.owl>

2 **Check the tests**

You can load a test suite from a URL, adding an RDF file that uses the VTC ontology.

Or you can copy the test suite code, if the test suite uses the VTC ontology.

Or you can add the tests directly. To add more than one test separate them by using ";".

The following link shows all the supported tests. In this other link you can also find some examples that can be useful to propose tests.

DIDEO_00000010 type Class

Something went wrong. Check that the ontology doesn't have any inconsistencies or unsatisfiable classes

Important information

- The test syntax is case sensitive (e.g., Sensor and sensor are detected as different terms)
- For generating the tests you have to use the [glossary of terms](#) (you can change it)
- The test syntax is available [here](#). You should insert the terms between brackets ("e.g., [Class] → Sensor"), while the italicised terms (e.g., type) are keywords that cannot be changed
- Terms in red indicate that they don't exist in the ontology, while the terms in orange indicate that the term exist in the ontology but the type (e.g., class or individual) is not the specified in the term
- The Absent relation result does not indicate that the test is not passed, but that something may be missing

Tests results

Test	Result	Problem
DIDEO_00000010 type Property	Undefined terms	The terms in the test are not correctly defined in the ontology

2

Check the tests

You can load a test suite from a URL, adding an RDF file that uses the VTC ontology.

Load from URI

Or you can copy the test suite code, if the test suite uses the VTC ontology.

Load from file

Or you can add the tests directly. To add more than one test separate them by using ";"

The following link shows all the supported tests. In this other link you can also find some examples that can be useful to propose tests.

Check

Something went wrong. Check that the ontology doesn't have any inconsistencies or unsatisfiable classes

Important information
















- The test syntax is case sensitive (e.g., Sensor and sensor are detected as different terms)
- For generating the tests you have to use the glossary of terms (you can change it)
- The test syntax is available here. You should insert the terms between brackets ("e.g., [Class] → Sensor"), while the italicised terms (e.g., type) are keywords that cannot be changed
- Terms in red indicate that they don't exist in the ontology, while the terms in orange indicate that the term exist in the ontology but the type (e.g., class or individual) is not the specified in the term
- The **Absent relation** result does not indicate that the test is not passed, but that something may be missing

Tests results

Test	Result	Problem
DIDEO_00000010 type Property	Undefined terms	The terms in the test are not correctly defined in the ontology

🗑️ Clean results

🔄 Export test suite

Tests results			
Test	Result	Problem	
HSO_0000314 disjointWith HSO_0000318	Undefined terms	The terms in the test are not defined in the ontology	
OBI_0000070 disjointWith OBI_0000339	Undefined terms	The terms in the test are not defined in the ontology	
BFO_0000016 disjointWith BFO_0000023	Undefined terms	The terms in the test are not defined in the ontology	
UBERON_0000913 subClassOf BFO_0000050 some UBERON_0000479	Undefined terms	The terms in the test are not defined in the ontology	
UBERON_0007779 subClassOf RO_0002494 some UBERON_0001969	Undefined terms	The terms in the test are not defined in the ontology	
UBERON_0000178 subClassOf BFO_0000051 some UBERON_0001969	Undefined terms	The terms in the test are not defined in the ontology	
SIO_000666 subClassOf ENVO_0100658	Undefined terms	The terms in the test are not defined in the ontology	
BFO_0000008 subClassOf BFO_0000003	Undefined terms	The terms in the test are not defined in the ontology	
BFO_0000006 subClassOf BFO_0000141	Undefined terms	The terms in the test are not defined in the ontology	
BSPO_0000108 type Property	Passed	None	
HSO_0000307 type Property	Passed	None	
HSO_0000376 type Property	Passed	None	
HSO_0000383 type Class	Undefined terms	The terms in the test are not defined in the ontology	
BFO_0000140 type Class	Undefined terms	The terms in the test are not defined in the ontology	
HSO_0000247 type Class	Undefined terms	The terms in the test are not defined in the ontology	

2 Check the tests

You can load a test suite from a URL, adding an RDF file that uses the VTC ontology.

Load from URI

Or you can copy the test suite code, if the test suite uses the VTC ontology.

Load from file

Or you can add the tests directly. To add more than one test separate them by using ";".

The following link shows all the supported tests. In this other link you can also find some examples that can be useful to propose tests.

BFO_0000055 range BFO_0000017

Check

Important information

- The test syntax is **case sensitive** (e.g., Sensor and sensor are detected as different terms)
- For generating the tests you have to use the **glossary of terms** (you can change it)
- The **test syntax** is available [here](#). You should insert the terms between brackets ("e.g., [Class] → Sensor"), while the *italised terms* (e.g., *type*) are keywords that cannot be changed
- Terms in **red** indicate that they don't exist in the ontology, while the terms in **orange** indicate that the term exist in the ontology but the type (e.g., class or individual) is not the specified in the term
- The **Absent relation** result does not indicate that the test is not passed, but that something may be missing

Tests results

Test	Result	Problem	
BFO_0000055 range BFO_0000017	Undefined terms	The terms in the test are not defined in the ontology	
HSO_0000301 domain HSO_0000001	Undefined terms	The terms in the test are not defined in the ontology	
GAZ_00002845 Type NCIT_C25464	Undefined terms	The terms in the test are not defined in the ontology	
GAZ_00002828 Type NCIT_C25464	Undefined terms	The terms in the test are not defined in the ontology	
GAZ_00002938 Type NCIT_C25464	Undefined terms	The terms in the test are not defined in the ontology	

<http://purl.obolibrary.org/obo/ido.owl>
[See the glossary of terms](#)
[Remove](#)

2 **Check the tests**

You can load a test suite from a URL, adding an RDF file that uses the VTC ontology.

[Load from URI](#)

Or you can copy the test suite code, if the test suite uses the VTC ontology.

[Load from file](#)

Or you can add the tests directly. To add more than one test separate them by using ";"

The [following link](#) shows all the supported tests. In [this other link](#) you can also find some examples that can be useful to propose tests.

BFO_0000144 **subClassOf** BFO_0000015 [Check](#)

Something went wrong. Check that the ontology doesn't have any inconsistencies or unsatisfiable classes

Important information

- The test syntax is **case sensitive** (e.g., Sensor and sensor are detected as different terms)
- For generating the tests you have to use the **glossary of terms** (you can change it)
- The test **syntax** is available [here](#). You should insert the terms between brackets ("e.g., [Class] → Sensor"), while the **italised terms** (e.g., *type*) are keywords that cannot be changed
- Terms in **red** indicate that they don't exist in the ontology, while the terms in **orange** indicate that the term exist in the ontology but the type (e.g., class or individual) is not the specified in the term
- The **Absent relation** result does not indicate that the test is not passed, but that something may be missing

Tests results

Test	Result	Problem	
RO_0000081 type Property	Passed	None	
RO_0002215 type Property	Passed	None	
RO_0001025 type Property	Passed	None	
IDO_0000594 type Class	Passed	None	
IDO_0000469 type Class	Passed	None	
IDO_0000484 type Class	Passed	None	

<http://purl.obolibrary.org/obo/ido.owl>
[See the glossary of terms](#)
[Remove](#)

2 **Check the tests**

You can load a test suite from a URL, adding an RDF file that uses the VTC ontology.

[Load from URI](#)

Or you can copy the test suite code, if the test suite uses the VTC ontology.

[Load from file](#)

Or you can add the tests directly. To add more than one test separate them by using ",".

The following link shows all the supported tests. In this other link you can also find some examples that can be useful to propose tests.

[Check](#)

Something went wrong. Check that the ontology doesn't have any inconsistencies or unsatisfiable classes

Important information

- The test syntax is case sensitive (e.g., Sensor and sensor are detected as different terms)
- For generating the tests you have to use the glossary of terms (you can change it)
- The test syntax is available [here](#). You should insert the terms between brackets ("e.g., [Class] → Sensor"), while the italicised terms (e.g., type) are keywords that cannot be changed
- Terms in red indicate that they don't exist in the ontology, while the terms in orange indicate that the term exist in the ontology but the type (e.g., class or individual) is not the specified in the term
- The **Absent relation** result does not indicate that the test is not passed, but that something may be missing

Tests results

Test	Result	Problem	
RO_0000081 type Property	Passed	None	
RO_0002215 type Property	Passed	None	
RO_0001025 type Property	Passed	None	
IDO_0000594 type Class	Passed	None	
IDO_0000469 type Class	Passed	None	
IDO_0000484 type Class	Passed	None	

2 Check the tests

You can load a test suite from a URL, adding an RDF file that uses the VTC ontology.

Load from URI

Or you can copy the test suite code, if the test suite uses the VTC ontology.

Load from file

Or you can add the tests directly. To add more than one test separate them by using ",".

The following link shows all the supported tests. In this other link you can also find some examples that can be useful to propose tests.

Check

Something went wrong. Check that the ontology doesn't have any inconsistencies or unsatisfiable classes

Important information

- The test syntax is **case sensitive** (e.g., Sensor and sensor are detected as different terms)
- For generating the tests you have to use the **glossary of terms** (you can change it)
- The **test syntax** is available [here](#). You should insert the terms between brackets ("e.g., [Class] → Sensor"), while the **italised terms** (e.g., *type*) are keywords that cannot be changed
- Terms in **red** indicate that they don't exist in the ontology, while the terms in **orange** indicate that the term exist in the ontology but the type (e.g., class or individual) is not the specified in the term
- The **Absent relation** result does not indicate that the test is not passed, but that something may be missing

Tests results

Test	Result	Problem	
RO_0000081 type Property	Passed	None	🗑
RO_0002215 type Property	Passed	None	🗑
RO_0001025 type Property	Passed	None	🗑
IDO_0000594 type Class	Passed	None	🗑
IDO_0000469 type Class	Passed	None	🗑
IDO_0000484 type Class	Passed	None	🗑

<http://purl.obolibrary.org/obo/MFOMD.owl>
See the glossary of terms
Remove

2 **Check the tests**
 You can load a test suite from a URL, adding an RDF file that uses the VTC ontology.

Load from URI

Or you can copy the test suite code, if the test suite uses the VTC ontology.

Load from file

Or you can add the tests directly. To add more than one test separate them by using ",".





The following link shows all the supported tests. In this other link you can also find some examples that can be useful to propose tests.

















Check

Important information













- The test syntax is **case sensitive** (e.g., Sensor and sensor are detected as different terms)
- For generating the tests you have to use the **glossary of terms** (you can change it)
- The **test syntax** is available [here](#). You should insert the terms between brackets ("e.g., [Class] → Sensor"), while the italicised terms (e.g., *type*) are keywords that cannot be changed
- Terms in **red** indicate that they don't exist in the ontology, while the terms in **orange** indicate that the term exist in the ontology but the type (e.g., class or individual) is not specified in the term
- The **Absent relation** result does not indicate that the test is not passed, but that something may be missing

Tests results











Test	Result	Problem	
BFO_0000062 domain BFO_0000003	Undefined terms	The terms in the test are not defined in the ontology	
BFO_0000054 range BFO_0000015	Absent relation	The ontology does not implement the requirement associated to the test	
IAO_0000227 type IAO_0000225	Passed	None	
IAO_0000124 type IAO_0000078	Passed	None	

IAO_0000410 type IAO_0000409	Passed	None	
BFO_0000002 disjointWith BFO_0000003	Passed	None	
BFO_0000016 disjointWith BFO_0000023	Passed	None	
BFO_0000008 disjointWith BFO_0000015	Passed	None	
MFOMD_0000041 subClassOf BFO_0000132 some MFOMD_0000025	Passed	None	
MF_0000024 subClassOf BFO_0000117 some MF_0000034	Passed	None	
MF_0000020 subClassOf BFO_0000117 some MF_0000017	Passed	None	
DOID_0060165 subClassOf DOID_8619	Passed	None	
DOID_8986 subClassOf DOID_535	Passed	None	
BFO_0000107 type Property	Passed	None	
BFO_0000117 type Property	Passed	None	
BFO_0000055 type Property	Passed	None	
MFOMD_0000001 type Class	Passed	None	
MFOMD_0000025 subClassOf OGMS_0000063	Passed	None	
MFOMD_0000041 type Class	Passed	None	
MFOMD_0000040 type Class	Passed	None	

Tests results

Test	Result	Problem	
BTO_0005672 subClassOf RO_0002202 some BTO_0001321	Passed	None	
BTO_0003089 subClassOf BFO_0000050 some BTO_0002098	Passed	None	
BTO_0000370 subClassOf BFO_0000050 some BTO_0000369	Passed	None	
BTO_0003622 subClassOf BTO_0000830	Passed	None	
BTO_0001255 subClassOf BTO_0000452	Passed	None	
BTO_0004468 subClassOf BTO_0000416	Passed	None	
BTO_0001492 type Class	Passed	None	
BTO_0002603 type Class	Passed	None	
BTO_0005352 type Class	Passed	None	
RO_0000053 type Property	Passed	None	
RO_0003001 type Property	Passed	None	
BFO_0000054 type Property	Passed	None	

Tests results

Test	Result	Problem	
CMO_0000704 subClassOf part_of some CMO_0000700	Absent relation	The ontology does not implement the requirement associated to the test	
CMO_0000047 subClassOf part_of some CMO_0002334	Absent relation	The ontology does not implement the requirement associated to the test	
CMO_0000395 subClassOf part_of some CMO_0001333	Absent relation	The ontology does not implement the requirement associated to the test	
CMO_0001707 subClassOf CMO_0001706	Passed	None	
CMO_0001756 subClassOf CMO_0002163	Passed	None	
CMO_0000456 subClassOf CMO_0000227	Passed	None	
CMO_0001818 type Class	Passed	None	
CMO_0000715 type Class	Passed	None	
CMO_0000714 type Class	Passed	None	
part_of type Property	Passed	None	

DOID testi ar Themis

Add the URI of the ontology to be validated

Add the code of the ontology to be validated

2 Check the tests
 You can load a test suite from a URL, add

Or you can copy the test suite code, if the test suite uses the VTC ontology.

Or you can add the tests directly. To add more than one test separate them by using ";".
 The following link shows all the supported tests. In this other link you can also find some examples that can be useful to propose tests.

```

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html> <head>
<title>502 Proxy Error</title>
</head> <body>
<h1>Proxy Error</h1>
<p>The proxy server received an invalid
response from an upstream server. <br />
The proxy server could not handle the request<p>Reason:
<strong>Error reading from remote server</strong></p></p>
</body> </html>

```

Important information

- The test syntax is case sensitive (e.g., Sensor and sensor are detected as different terms)
- For generating the tests you have to use the glossary of terms (you can change it)
- The test syntax is available [here](#). You should insert the terms between brackets ("e.g., [Class] → Sensor"), while the italicised terms (e.g., type) are keywords that cannot be changed
- Terms in **red** indicate that they don't exist in the ontology, while the terms in **orange** indicate that the term exist in the ontology but the type (e.g., class or individual) is not the specified in the term
- The **Absent relation** result does not indicate that the test is not passed, but that something may be missing

ECO testi ar TDDOnto

Active ontology: Entities, Individuals by class, DL Query

Ontology header: Metrics, TDDOnto

Ontology IRI: http://purl.obolibrary.org/obo/eco.owl

Ontology Version IRI: http://purl.obolibrary.org/obo/eco/releases/2022-08-05/eco.owl

title: http://purl.obolibrary.org/obo/eco/releases/2022-08-05/eco.owl

Evidence & Conclusion Ontology (ECO)

description: The Evidence & Conclusion Ontology (ECO) describes types of scientific evidence within the biological research domain that arise from laboratory experiments, computational methods, literature curation, or other means.

license: <https://creativecommons.org/publicdomain/zero/1.0/>

rdfs:comment: ECO is a public domain ontology released into the public domain under CC0 1.0 Universal (CC0 1.0). Anyone is free to copy, modify, or distribute the work, even for commercial purposes, without asking permission. Please see the Public Domain Dedication (<https://creativecommons.org/publicdomain/zero/1.0/>) for an easy-to-read description of CC0 1.0 or the full legal code (<https://creativecommons.org/publicdomain/zero/1.0/>) for more detailed information. To get a sense of any ECO in CC0 as opposed to licensed under CC-BY, please read this thoughtful discussion (<https://github.com/obolibrary/obolibrary.github.io/releases/2022-08-05/eco>) on the OBO Foundry GitHub site.

date: 05-08-2022 15:33

default: namespace: eco

New test

Tests

ECO_0007234 type Class	Undefined	<input type="checkbox"/>
ECO_0007742 type Class	Undefined	<input type="checkbox"/>
ECO_0001809 type Class	Undefined	<input type="checkbox"/>
ECO_0000001 type Property	Undefined	<input type="checkbox"/>
ECO_0000000 type Property	Undefined	<input type="checkbox"/>
IAO_0002222 type Property	Undefined	<input type="checkbox"/>
ECO_0007225 subClassOf ECO_0000180	OK	<input type="checkbox"/>
ECO_0001983 subClassOf ECO_0001982	OK	<input type="checkbox"/>
ECO_0006376 subClassOf ECO_0006343	OK	<input type="checkbox"/>
ECO_0000086 subClassOf IAO_0000136 some GO_0000306	OK	<input type="checkbox"/>
ECO_0000170 subClassOf IAO_0000136 some GO_0001046	OK	<input type="checkbox"/>
ECO_0000000 disjointWith ECO_0000217	OK	<input type="checkbox"/>
ECO_0000023 disjointWith ECO_0000218	OK	<input type="checkbox"/>
ECO_0000033 disjointWith ECO_0000034	OK	<input type="checkbox"/>
IAO_0000410 type IAO_0000409	Undefined	<input type="checkbox"/>
IAO_0000226 type IAO_0000226	Undefined	<input type="checkbox"/>
IAO_0000124 type IAO_0000078	Undefined	<input type="checkbox"/>
BFO_0000052 domain BFO_0000003	Undefined	<input type="checkbox"/>
BFO_0000054 range BFO_0000015	Undefined	<input type="checkbox"/>
ECO_0005605 SubClassOf IAO_0000136 some GO_0045677	OK	<input checked="" type="checkbox"/>

Ontology imports: Ontology Prefixes: General class axioms

GENO testi ar TDDOnto

Active ontology: Entires > Individuals by class > DL Query

Ontology header: <http://purl.obolibrary.org/obo/geno.owl>

Ontology Version IRI: <http://purl.obolibrary.org/obo/geno/releases/2022-08-10/geno.owl>

Annotations:

- dc:title: GENO ontology
- dc:description: GENO is an OWL model of genotypes, their more fundamental sequence components, and lines to related biological and experimental entities. It assesses many parts of the model as exploratory and set to undergo refactoring. In addition, many classes and properties have GENO IRI's but are place holders for classes that will be imported from an external ontology (e.g. SO, CHEBI, OBI, etc). Furthermore, ongoing work will implement a model of genotype-phenotype associations. This will support description of assisted and ordered relationships between a genotypes, phenotypes, and environments, and the evidence/provenance behind these associations. Documentation is under development as well, and for now a slideshock is available at <http://www.slideshare.net/mh1209/vh-obo-2013>
- dc:license: <https://creativecommons.org/licenses/by/4.0/>
- owl:versionInfo: type: owl:Thing
- 2022-08-10

Tests:

Test	Status	Checkmark
GENO_0000229 Range GENO_0000702	Undefined	✓
GENO_0000221 Type Class	Undefined	✓
GENO_0000702 Type Class	Undefined	✓
GENO_0000222 Type Class	Undefined	✓
IAO_0000219 type Property	Undefined	✓
IAO_0000424 Type Property	Undefined	✓
IAO_0000425 Type Property	Undefined	✓
GENO_0000895 subClassOf GENO_0000823	OK	✓
GENO_0000823 subClassOf GENO_0000897	OK	✓
GENO_0000811 subClassOf GENO_0000899	OK	✓
GENO_0000827 subClassOf IAO_0000219 some GENO_0000516	OK	✓
GENO_0000802 subClassOf GENO_0000583 some GENO_0000336	OK	✓
GENO_0000722 subClassOf GENO_0000783 some GENO_0000782	OK	✓
BFO_0000022 disjointWith BFO_0000003	Undefined	✓
BFO_0000019 disjointWith BFO_0000017	OK	✓
GENO_0000015 disjointWith GENO_0000702	OK	✓
GENO_0000484 Type IAO_0000102	Undefined	✓
IAO_0000410 Type IAO_0000409	Undefined	✓
IAO_0000420 Type IAO_0000409	Undefined	✓
GENO_0000856 Domain GENO_0000815	Undefined	✓

Execute tests Delete selected tests Add to ontology Select / Unselect All

DIDEO testi ar TDDOnto

Active ontology: Entires > Individuals by class > DL Query

Ontology header: <http://purl.obolibrary.org/obo/dideo.owl>

Ontology Version IRI: <http://purl.obolibrary.org/obo/dideo/releases/2022-06-14/dideo.owl>

Annotations:

- label: [language: en] DIDEO
- dc:creator: Mathias Brochhausen
- dc:contributor: Anuj Shah
- dc:contributor: Dan Malone
- dc:contributor: Jessica Tay-Suntheimer
- dc:contributor: Jingling Yu
- dc:contributor: Jodi Schneider
- dc:contributor: Philip E. Emsay
- dc:contributor: Richard D. Boyce
- dc:contributor: William R. Hogan
- rdft:comment: [language: en] The Drug-Drug Interaction and Drug-Drug Interaction Evidence Ontology (DIDEO) by the DIDEO development group is licensed under CC BY 4.0 <https://creativecommons.org/licenses/by/4.0/>

Tests:

Test	Status	Checkmark
BFO_0000057 range BFO_0000002	Undefined	✓
DIDEO_00000140 type Class	Undefined	✓
DIDEO_00000162 type Class	Undefined	✓
DIDEO_00000102 type Class	Undefined	✓
DIDEO_0000027 type Property	Undefined	✓
DIDEO_0000098 type Property	Undefined	✓
DIDEO_0000090 type Property	Undefined	✓
DIDEO_0000034 subClassOf BFO_0000016	OK	✓
DIDEO_00000173 subClassOf BFO_0000090	OK	✓
DIDEO_0000189 subClassOf CL_0000010	OK	✓
DIDEO_0000019 subClassOf BFO_0000052 some DRONL_0000005	OK	✓
DIDEO_0000039 subClassOf BFO_0000052 some OBI_0000427	OK	✓
DIDEO_0000020 subClassOf BFO_0000057 some BFO_0000040	OK	✓
BFO_0000016 disjointWith BFO_0000023	OK	✓
BFO_0000017 disjointWith BFO_0000019	Undefined	✓
BFO_0000004 disjointWith BFO_0000031	Undefined	✓
OBI_0302900 type OBI_0302900	Undefined	✓
IAO_0000124 type IAO_0000078	Undefined	✓
IAO_0000229 type IAO_0000225	Undefined	✓
IAO_0000136 domain IAO_0000030	Undefined	✓

Execute tests Delete selected tests Add to ontology Select / Unselect All

HSD testi ar TDDOnto

Active ontology: Entires > Individuals by class > DL Query

Ontology header: <http://purl.obolibrary.org/obo/hso.owl>

Ontology Version IRI: <http://purl.obolibrary.org/obo/hso/2021-12-13/hso.owl>

Annotations:

- rdfs:comment: [language: en] This is an ontology to support (one) health surveillance, focused on "surveillance system level data", that is, data outputs from surveillance activities, such as number of samples collected, cases observed, etc.
- dc:license: <https://creativecommons.org/licenses/by/4.0/>
- dc:creator: Fernanda Diersa
- dc:description: [language: en] An ontology to support semantic interoperability among domains involved in health surveillance (animal health, public health and food safety). More details at <https://github.com/healthsurveillance.org>
- dc:terms:title: [language: en] Health Surveillance Ontology (HSD)
- foaf:homepage: <https://h3d.org/hso>

Tests:

Test	Status	Checkmark
BFO_0000055 Range BFO_0000017	Undefined	✓
BFO_0000006 subClassOf BFO_0000141	OK	✓
BFO_0000008 subClassOf BFO_0000003	OK	✓
SIO_000066 subClassOf EXNO_01000658	OK	✓
UBERONL_0000178 subClassOf BFO_0000051 some UBERONL_0001969	OK	✓
UBERONL_0000779 subClassOf RO_0002484 some UBERONL_0001969	OK	✓
UBERONL_0000913 subClassOf BFO_0000050 some UBERONL_0000479	OK	✓
BFO_0000019 disjointWith BFO_0000023	OK	✓
CIL_0000078 disjointWith CIL_0000339	OK	✓
HSD_0000314 disjointWith HSD_0000318	OK	✓
HSD_0000383 type Class	Undefined	✓
BFO_0000140 type Class	Undefined	✓
HSD_0000247 type Class	Undefined	✓
HSD_0000379 type Property	Undefined	✓
HSD_0000303 type Property	Undefined	✓
BSPPO_0000198 type Property	Undefined	✓
GAZ_00002938 Type NCT_C25464	Undefined	✓
GAZ_00002939 Type NCT_C25464	Undefined	✓
GAZ_00002845 Type NCT_C25464	Undefined	✓
HSD_0000391 domain HSD_0000001	Undefined	✓
BFO_0000055 Range BFO_0000017	Undefined	✓

Execute tests Delete selected tests Add to ontology Select / Unselect All

MFOMD testi ar TDDOnto

Active ontology: MFOMD (http://purl.obolibrary.org/obo/MFOMD/2020-04-26/MFOMD.owl)

Ontology IRI: http://purl.obolibrary.org/obo/MFOMD.owl

Ontology Version IRI: http://purl.obolibrary.org/obo/MFOMD/2020-04-26/MFOMD.owl

Annotations:

- dc:title: Mental Disease Ontology
- dc:description: The Mental Disease Ontology is developed to facilitate representation for all aspects of mental diseases. It is an extension of the Ontology for General Medical Science (OGMS) and Mental Functioning Ontology (MFO).
- dc:contributor: Barry Smith, Janna Hastings, Joaquina Alicia Chacón Candia, Mark Jensen
- http://www.semanticweb.org/ontology/11/1.0
- owl:versionInfo: 2020-04-26

New test: BFO_000054 Range BFO_000015

Tests:

Test	Status
MFOMD_000040 type Class	Undefined
MFOMD_000041 type Class	Undefined
MFOMD_000001 type Class	Undefined
BFO_000055 type Property	Undefined
BFO_000117 type Property	Undefined
BFO_000107 type Property	Undefined
MFOMD_000025 subclassOf OGMS_0000063	OK
DOID_0986 subclassOf DOID_535	Undefined
DOID_006165 subclassOf DOID_0619	Undefined
MF_000020 subclassOf BFO_000117 some MF_000017	Undefined
MF_000024 subclassOf BFO_000117 some MF_000034	Undefined
MFOMD_000041 subclassOf BFO_000152 some MFOMD_000025	Undefined
BFO_000008 disjointWith BFO_000015	Undefined
BFO_000016 disjointWith BFO_000023	Undefined
BFO_000002 disjointWith BFO_000093	Undefined
IAO_0000410 type IAO_0000409	Undefined
IAO_000124 type IAO_0000078	Undefined
IAO_000027 type IAO_0000225	Undefined
BFO_000062 domain BFO_000003	Undefined
BFO_000054 range BFO_000015	Undefined

BTO testi ar TDDOnto

Active ontology: bto (http://purl.obolibrary.org/obo/bto/releases/2021-10-25/bto.owl)

Ontology IRI: http://purl.obolibrary.org/obo/bto

Ontology Version IRI: http://purl.obolibrary.org/obo/bto/releases

Annotations:

- dc:title: The BRENDA Tissue Ontology (BTO)
- dc:description: A structured controlled vocabulary for the source of an enzyme comprising tissues, cell lines, cell types and cell cultures.
- auto-generated-by: OBO-Edit 2.3.1
- date: 23.10.2021 16.56
- default_namespace: BrendaTissuuOBO
- has_obo_format_version: 1.2
- saved-by: Marion

New test: BTO_0005672 SubClassOf RO_0002202 some BTO_0001321

Tests:

Test	Status
BTO_0005362 type Class	Undefined
BTO_0002603 type Class	Undefined
BTO_0001482 type Class	Undefined
BFO_000054 type Property	Undefined
RO_0003001 type Property	Undefined
RO_000053 type Property	Undefined
BTO_0004469 subclassOf BTO_0000416	OK
BTO_0001255 subclassOf BTO_0000452	OK
BTO_0003622 subclassOf BTO_0000830	OK
BTO_0000370 subclassOf BFO_0000050 some BTO_0000369	OK
BTO_0003089 subclassOf BFO_0000050 some BTO_0002098	OK
BTO_0005672 subclassOf RO_0002202 some BTO_0001321	OK

CMO testi ar TDDOnto

Active ontology: cmo obo (http://purl.obolibrary.org/obo/cmo/cmo/releases/2019-02-19/cmo.owl/cmo.obo.owl)

Ontology IRI: http://purl.obolibrary.org/obo/cmo.obo.owl

Ontology Version IRI: http://purl.obolibrary.org/obo/cmo/releases/2019-02-19/cmo.owl/cmo.obo.owl

Annotations:

- has_obo_format_version: 1.2

New test: CMO_0000704 SubClassOf part_of some CMO_0000700

Tests:

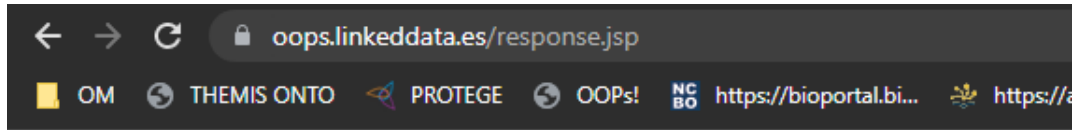
Test	Status
CMO_0000714 Type Class	Undefined
CMO_0000715 Type Class	Undefined
CMO_0001818 Type Class	Undefined
part_of Type Property	Undefined
CMO_0000456 SubClassOf CMO_0000227	OK
CMO_0001796 SubClassOf CMO_0002163	OK
CMO_0001797 SubClassOf CMO_0001796	OK
CMO_0000395 SubClassOf part_of some CMO_0001333	OK
CMO_0000847 SubClassOf part_of some CMO_0002334	OK
CMO_0000704 SubClassOf part_of some CMO_0000700	OK

DOID testi ar TDDOnto

The screenshot shows the DOID ontology interface. The left pane displays a class hierarchy for 'lymphoma', listing various cancer types such as cervical cancer, cheek mucosa cancer, choroid cancer, and lymphoma. The right pane shows the 'Annotations' for 'lymphoma', including labels, IDs, and alternative IDs. A 'Description' dialog box is open, displaying an error message: 'An error occurred during reasoning'. Below the dialog, the 'SubClass Of' section lists several axioms, including 'disease has location some lymphoid tissue' and 'hematologic cancer'.

OBI testi ar TDDOnto

The screenshot shows the OBI ontology interface. The left pane displays the 'Ontology IRI' and 'Ontology Version IRI'. The right pane shows the 'TDDOnto' section, which includes a 'New test' form and a list of tests. The tests are organized into a table with columns for 'Test ID', 'Test Name', and 'Status'. The tests include 'OBI_0000097 Range BFO_0000023' and several other tests with various statuses like 'Undefined', 'OK', and 'Error'.



Proxy Error

The proxy server received an invalid response from an upstream server.
The proxy server could not handle the request

Reason: **Error reading from remote server**

Apache Server at oops.linkeddata.es Port 443

GENO testi ar OOPs!

OOPSI (Ontology Pitfall Scanner!) helps you to detect some of the most common pitfalls appearing when developing ontologies. To try it, enter a URI or paste an OWL document into the text field above. A list of pitfalls and the elements of your ontology where they appear will be displayed.

Scanner by URI:

Example: http://oops.linkeddata.es/example/swc_2009-05-09.rdf

Scanner by direct input:

Uncheck this checkbox if you don't want us to keep a copy of your ontology. [Go to advanced evaluation](#)

Evaluation results

It is obvious that not all the pitfalls are equally important; their impact in the ontology will depend on multiple factors. For this reason, each pitfall has an importance level attached indicating how important it is. We have identified three levels:

- Critical** (red circle): It is crucial to correct the pitfall. Otherwise, it could affect the ontology consistency, reasoning, applicability, etc.
- Important** (orange circle): Though not critical for ontology function, it is important to correct this type of pitfall.
- Minor** (yellow circle): It is not really a problem, but by correcting it we will make the ontology nicer.

[Expand All] | [Collapse All]

Results for P04: Creating unconnected ontology elements.	1 case Minor
Results for P08: Missing annotations.	1 case Minor
Results for P11: Missing domain or range in properties.	130 cases Important
Results for P13: Inverse relationships not explicitly declared.	86 cases Minor
Results for P20: Misusing ontology annotations.	1 case Minor
Results for P22: Using different naming conventions in the ontology.	ontology* Minor
Results for P24: Using recursive definitions.	1 case Important
Results for P30: Equivalent classes not explicitly declared.	1 case Important
Results for P32: Several classes with the same label.	4 cases Minor
Results for P36: URI contains file extension.	ontology* Minor
Results for P39: Ambiguous namespace.	ontology* Critical
Results for P40: Namespace hijacking.	1 case Critical
Results for P41: No license declared.	ontology* Important

Want to help?

- Suggest new pitfalls
- Provide feedback

Documentation:

- Pitfall catalogue
- User guide
- Technical report

Related papers:

- ISWIS 2014
- EKAW 2012
- ESWC 2012 Demo
- Ontoqual 2010
- CAEPIA 2009

Web services:

- REST Web Service

Developed by:

DIDEO testi ar OOPs!

Scanner by URI: Scanner by URI

Example: http://oops.linkeddata.es/example/swc_2009-05-09.rdf

If you just include the RDF code here, the following Pitfalls will not be checked:
P36. URI contains file extension, P37. Ontology not available, P40. Namespace hijacking

Scanner by direct input: Scanner by RDF

Uncheck this checkbox if you don't want us to keep a copy of your ontology. [Go to advanced evaluation](#)

Evaluation results

It is obvious that not all the pitfalls are equally important; their impact in the ontology will depend on multiple factors. For this reason, each pitfall has an importance level attached indicating how important it is. We have identified three levels:

- **Critical** 🚫 : It is crucial to correct the pitfall. Otherwise, it could affect the ontology consistency, reasoning, applicability, etc.
- **Important** ⚠️ : Though not critical for ontology function, it is important to correct this type of pitfall.
- **Minor** 🟡 : It is not really a problem, but by correcting it we will make the ontology nicer.

[Expand All] | [Collapse All]

Results for P04: Creating unconnected ontology elements.	1 case Minor 🟡
Results for P08: Missing annotations.	6 cases Minor 🟡
Results for P11: Missing domain or range in properties.	81 cases Important ⚠️
Results for P13: Inverse relationships not explicitly declared.	58 cases Minor 🟡
Results for P19: Defining multiple domains or ranges in properties.	1 case Critical 🚫
Results for P20: Misusing ontology annotations.	4 cases Minor 🟡
Results for P22: Using different naming conventions in the ontology.	ontology* Minor 🟡
Results for P29: Defining wrong transitive relationships.	1 case Critical 🚫
Results for P30: Equivalent classes not explicitly declared.	2 cases Important ⚠️
Results for P32: Several classes with the same label.	5 cases Minor 🟡
Results for P34: Untyped class.	5 cases Important ⚠️
Results for P36: URI contains file extension.	ontology* Minor 🟡
Results for P39: Ambiguous namespace.	ontology* Critical 🚫
Results for P41: No license declared.	ontology* Important ⚠️
SUGGESTION: symmetric or transitive object properties.	3 cases

Want to help?

- Suggest new pitfalls
- Provide feedback

Documentation:

- Pitfall catalogue
- User guide
- Technical report

Related papers:

- IJSWIS 2014
- EKAW 2012
- ESWC 2012 Demo
- Ontoqual 2010
- CAEPIA 2009

Web services:

- REST Web Service

Developed by:



HSO testi ar OOPs!

OOPSI! (Ontology Pitfall Scanner!) helps you to detect some of the most common pitfalls appearing when developing ontologies. To try it, enter a URI or paste an OWL document into the text field above. A list of pitfalls and the elements of your ontology where they appear will be displayed.

Scanner by URI:

Example: http://oops.linkeddata.es/example/swc_2009-05-09.rdf

Scanner by direct input:

Uncheck this checkbox if you don't want us to keep a copy of your ontology. [Go to advanced evaluation](#)

Evaluation results

It is obvious that not all the pitfalls are equally important; their impact in the ontology will depend on multiple factors. For this reason, each pitfall has an importance level attached indicating how important it is. We have identified three levels:

- **Critical** 🚫 : It is crucial to correct the pitfall. Otherwise, it could affect the ontology consistency, reasoning, applicability, etc.
- **Important** ⚠️ : Though not critical for ontology function, it is important to correct this type of pitfall.
- **Minor** 🟡 : It is not really a problem, but by correcting it we will make the ontology nicer.

[Expand All] | [Collapse All]

Results for P04: Creating unconnected ontology elements.	1 case Minor 🟡
Results for P08: Missing annotations.	9 cases Minor 🟡
Results for P11: Missing domain or range in properties.	103 cases Important ⚠️
Results for P13: Inverse relationships not explicitly declared.	103 cases Minor 🟡
Results for P19: Defining multiple domains or ranges in properties.	1 case Critical 🚫
Results for P22: Using different naming conventions in the ontology.	ontology* Minor 🟡
Results for P30: Equivalent classes not explicitly declared.	1 case Important ⚠️
Results for P32: Several classes with the same label.	4 cases Minor 🟡
Results for P36: URI contains file extension.	ontology* Minor 🟡
Results for P39: Ambiguous namespace.	ontology* Critical 🚫
Results for P40: Namespace hijacking.	11 cases Critical 🚫
Results for P41: No license declared.	ontology* Important ⚠️
SUGGESTION: symmetric or transitive object properties.	3 cases

Want to help?

- Suggest new pitfalls
- Provide feedback

Documentation:

- Pitfall catalogue
- User guide
- Technical report

Related papers:

- IJSWIS 2014
- EKAW 2012
- ESWC 2012 Demo
- Ontoqual 2010
- CAEPIA 2009

Web services:

- REST Web Service

Developed by:

OOPS! (Ontology Pitfall Scanner!) helps you to detect some of the most common pitfalls appearing when developing ontologies. To try it, enter a URI or paste an OWL document into the text field above. A list of pitfalls and the elements of your ontology where they appear will be displayed.

Scanner by URI:
 Example: http://oops.linkeddata.es/example/swc_2009-05-09.rdf

Scanner by direct input:

Uncheck this checkbox if you don't want us to keep a copy of your ontology.

[Go to advanced evaluation](#)

Evaluation results

It is obvious that not all the pitfalls are equally important; their impact in the ontology will depend on multiple factors. For this reason, each pitfall has an importance level attached indicating how important it is. We have identified three levels:

- **Critical** 🚫: It is crucial to correct the pitfall. Otherwise, it could affect the ontology consistency, reasoning, applicability, etc.
- **Important** ⚠️: Though not critical for ontology function, it is important to correct this type of pitfall.
- **Minor** 🟡: It is not really a problem, but by correcting it we will make the ontology nicer.

[\[Expand All\]](#) | [\[Collapse All\]](#)

Results for P04: Creating unconnected ontology elements.	2 cases Minor 🟡
Results for P08: Missing annotations.	5 cases Minor 🟡
Results for P11: Missing domain or range in properties.	23 cases Important ⚠️
Results for P13: Inverse relationships not explicitly declared.	7 cases Minor 🟡
Results for P22: Using different naming conventions in the ontology.	ontology* Minor 🟡
Results for P30: Equivalent classes not explicitly declared.	2 cases Important ⚠️
Results for P36: URI contains file extension.	ontology* Minor 🟡
Results for P39: Ambiguous namespace.	ontology* Critical 🚫
Results for P40: Namespace hijacking.	1 case Critical 🚫
Results for P41: No license declared.	ontology* Important ⚠️

Want to help?

- [Suggest new pitfalls](#)
- [Provide feedback](#)

Documentation:

- [Pitfall catalogue](#)
- [User guide](#)
- [Technical report](#)

Related papers:

- [ISWIS 2014](#)
- [EKAW 2012](#)
- [ESWC 2012 Demo](#)
- [Ontoqual 2010](#)
- [CAEPIA 2009](#)

Web services:

MFOMD testi ar OOPs!

Scanner by URI:

Example: http://oops.linkeddata.es/example/swc_2009-05-09.rdf

Scanner by direct input:

Uncheck this checkbox if you don't want us to keep a copy of your ontology. [Go to advanced evaluation](#)

Evaluation results

It is obvious that not all the pitfalls are equally important; their impact in the ontology will depend on multiple factors. For this reason, each pitfall has an importance level attached indicating how important it is. We have identified three levels:

- **Critical** 🚫 : It is crucial to correct the pitfall. Otherwise, it could affect the ontology consistency, reasoning, applicability, etc.
- **Important** ⚠️ : Though not critical for ontology function, it is important to correct this type of pitfall.
- **Minor** 🟡 : It is not really a problem, but by correcting it we will make the ontology nicer.

[Expand All] | [Collapse All]

Results for P03: Creating the relationship "is" instead of using "rdfs:subClassOf", "rdf:type" or "owl:sameAs".	1 case Critical 🚫
Results for P04: Creating unconnected ontology elements.	2 cases Minor 🟡
Results for P08: Missing annotations.	18 cases Minor 🟡
Results for P11: Missing domain or range in properties.	41 cases Important ⚠️
Results for P12: Equivalent properties not explicitly declared.	2 cases Important ⚠️
Results for P13: Inverse relationships not explicitly declared.	37 cases Minor 🟡
Results for P20: Misusing ontology annotations.	4 cases Minor 🟡
Results for P22: Using different naming conventions in the ontology.	ontology* Minor 🟡
Results for P24: Using recursive definitions.	1 case Important ⚠️
Results for P30: Equivalent classes not explicitly declared.	7 cases Important ⚠️
Results for P32: Several classes with the same label.	1 case Minor 🟡
Results for P36: URI contains file extension.	ontology* Minor 🟡
Results for P39: Ambiguous namespace.	ontology* Critical 🚫
Results for P41: No license declared.	ontology* Important ⚠️

Want to help?

- Suggest new pitfalls
- Provide feedback

Documentation:

- Pitfall catalogue
- User guide
- Technical report

Related papers:

- IJSWIS 2014
- EKAW 2012
- ESWC 2012 Demo
- Ontoqual 2010
- CAEPIA 2009

Web services:

- REST Web Service

Developed by:



CMO testi ar OOPs!

OOPSI! (Ontology Pitfall Scanner!) helps you to detect some of the most common pitfalls appearing when developing ontologies. To try it, enter a URI or paste an OWL document into the text field above. A list of pitfalls and the elements of your ontology where they appear will be displayed.

Scanner by URI:

Example: http://oops.linkeddata.es/example/swc_2009-05-09.rdf

Scanner by direct input:

Uncheck this checkbox if you don't want us to keep a copy of your ontology. [Go to advanced evaluation](#)

Evaluation results

It is obvious that not all the pitfalls are equally important; their impact in the ontology will depend on multiple factors. For this reason, each pitfall has an importance level attached indicating how important it is. We have identified three levels:

- **Critical** 🚫 : It is crucial to correct the pitfall. Otherwise, it could affect the ontology consistency, reasoning, applicability, etc.
- **Important** ⚠️ : Though not critical for ontology function, it is important to correct this type of pitfall.
- **Minor** 🟡 : It is not really a problem, but by correcting it we will make the ontology nicer.

[Expand All] | [Collapse All]

Results for P37: Ontology not available on the Web.	ontology* Critical 🚫
Results for P38: No OWL ontology declaration.	ontology* Important ⚠️
Results for P39: Ambiguous namespace.	ontology* Critical 🚫

Want to help?

- Suggest new pitfalls
- Provide feedback

Documentation:

- Pitfall catalogue
- User guide
- Technical report

DOID testi ar OOPs!

OOPS! (Ontology Pitfall Scanner!) helps you to detect some of the most common pitfalls appearing when developing ontologies. To try it, enter a URI or paste an OWL document into the text field above. A list of pitfalls and the elements of your ontology where they appear will be displayed.

Scanner by URI:
Example: http://oops.linkeddata.es/example/swc_2009-05-09.rdf

Scanner by direct input:

Uncheck this checkbox if you don't want us to keep a copy of your ontology. [Go to advanced evaluation](#)

Evaluation results

It is obvious that not all the pitfalls are equally important; their impact in the ontology will depend on multiple factors. For this reason, each pitfall has an importance level attached indicating how important it is. We have identified three levels:

- **Critical** 🚫 : It is crucial to correct the pitfall. Otherwise, it could affect the ontology consistency, reasoning, applicability, etc.
- **Important** ⚠️ : Though not critical for ontology function, it is important to correct this type of pitfall.
- **Minor** 🟡 : It is not really a problem, but by correcting it we will make the ontology nicer.

[Expand All] | [Collapse All]

Results for P36: URI contains file extension. ontology* | Minor 🟡

According to the highest importance level of pitfall found in your ontology the conformace badge suggested is "Minor pitfalls" (see below). You can use the following HTML code to insert the badge within your ontology documentation:

Want to help?

- Suggest new pitfalls
- Provide feedback

Documentation:

- Pitfall catalogue
- User guide
- Technical report

BTO testi ar Protege

< > **bto** (<http://purl.obolibrary.org/obo/bto/releases/2021-10-26/bto.owl>)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?subject ?label
WHERE { ?subject a owl:Class .
OPTIONAL { ?subject rdfs:label ?label } .
FILTER regex( ?label , "1-LN", "i" )
} ORDER BY ?subject
```

subject | label

bto (http://purl.obolibrary.org/obo/bto/releases/2021-10-26/bto.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?subject ?label
WHERE { ?subject a owl:Class .
OPTIONAL { ?subject rdfs:label ?label } .
FILTER regex( ?label , "11-9", "i" )
} ORDER BY ?subject

```

subject	label
---------	-------

bto (http://purl.obolibrary.org/obo/bto/releases/2021-10-26/bto.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?subject ?label
WHERE { ?subject a owl:Class .
OPTIONAL { ?subject rdfs:label ?label } .
FILTER regex( ?label , "Limb", "i" )
} ORDER BY ?subject

```

subject	label
CMO_0000019	"proximal hind limb c
CMO_0000185	"limb morphological r
CMO_0000186	"distal hind limb circ
CMO_0000187	"distal forelimb circur
CMO_0000188	"forelimb length"^^<
CMO_0000189	"hind limb length"^^<
CMO_0000317	"proximal forelimb ci
CMO_0000387	"forelimb morphologi
CMO_0000388	"hind limb morpholog
CMO_0000424	"forelimb weight"^^<
CMO_0001494	"hind limb weight"^^<

bto (http://purl.obolibrary.org/obo/bto/releases/2021-10-26/bto.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?label
WHERE {
  ?x rdf:type owl:ObjectProperty
  OPTIONAL { ?x rdfs:label ?label } .
  FILTER regex( ?label , "Realized", "i" )
}
```

x label

bto (http://purl.obolibrary.org/obo/bto/releases/2021-10-26/bto.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?label
WHERE {
  ?x rdf:type owl:ObjectProperty
  OPTIONAL { ?x rdfs:label ?label } .
  FILTER regex( ?label , "Prod", "i" )
}
```

x label

bto (http://purl.obolibrary.org/obo/bto/releases/2021-10-26/bto.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?label
WHERE {
  ?x rdf:type owl:ObjectProperty
  OPTIONAL { ?x rdfs:label ?label } .
  FILTER regex( ?label , "Bear", "i" )
}

```

x label

bto (http://purl.obolibrary.org/obo/bto/releases/2021-10-26/bto.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x rdfs:subClassOf ?y .
  OPTIONAL { ?x rdfs:label ?label }
  FILTER regex( ?label , "Lip e", "i" )
} ORDER BY ?x

```

x y

bto (http://purl.obolibrary.org/obo/bto/releases/2021-10-26/bto.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x rdfs:subClassOf ?y .
  OPTIONAL { ?x rdfs:label ?label }
  FILTER regex( ?label , "Skin fib", "i" )
} ORDER BY ?x
```

x y

bto (http://purl.obolibrary.org/obo/bto/releases/2021-10-26/bto.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x rdfs:subClassOf ?y .
  OPTIONAL { ?x rdfs:label ?label }
  FILTER regex( ?label , "Mucosal mast cell", "i" )
} ORDER BY ?x
```

x y

[bto \(http://purl.obolibrary.org/obo/bto/releases/2021-10-26/bto.owl\)](#)

[Active ontology](#) x [Entities](#) x [Individuals by class](#) x [DL Query](#) x [SPARQL Query](#) x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x rdfs:subClassOf ?y .
  OPTIONAL { ?x rdfs:label ?label }
  FILTER regex( ?label , "Egg white", "i" )
} ORDER BY ?x
  
```

x y

[bto \(http://purl.obolibrary.org/obo/bto/releases/2021-10-26/bto.owl\)](#)

[Active ontology](#) x [Entities](#) x [Individuals by class](#) x [DL Query](#) x [SPARQL Query](#) x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x rdfs:subClassOf ?y .
  OPTIONAL { ?x rdfs:label ?label }
  FILTER regex( ?label , "Promontor", "i" )
} ORDER BY ?x
  
```

x y

bto (<http://purl.obolibrary.org/obo/bto/releases/2021-10-26/bto.owl>)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x rdfs:subClassOf ?y .
  OPTIONAL { ?x rdfs:label ?label }
  FILTER regex( ?label , "LNCaP", "i" )
} ORDER BY ?x

```

x	y
---	---

CMO testi ar Protege

cmo (<http://purl.obolibrary.org/obo/cmo/releases/2019-02-19/cmo.owl>)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?subject ?label
WHERE {
  ?subject a owl:Class .
  OPTIONAL { ?subject rdfs:label ?label } .
  FILTER regex( ?label , "Calculated weight of islet", "i" )
} ORDER BY ?subject

```

subject	label
calculated weight of islet beta cells in duodenal region of pancreas	"calculated weight of islet beta cells in duodenal region of pancreas"^^xsd:string

cmo (<http://purl.obolibrary.org/obo/cmo/releases/2019-02-19/cmo.owl>)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?label
WHERE {
  ?x rdf:type owl:ObjectProperty
  OPTIONAL { ?x rdfs:label ?label } .
}

```

x	label
part_of	"part_of"^^<http://www.w3.org/2001/XMLSchema#string>

[cmo](#) (<http://purl.obolibrary.org/obo/cmo/releases/2019-02-19/cmo.owl>)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x rdfs:subClassOf ?y .
  OPTIONAL { ?x rdfs:label ?label }
  FILTER regex(?label, "Adipocyte", "i")
} ORDER BY ?x
  
```

x	y
adipocyte measurement	cell measurement

[cmo](#) (<http://purl.obolibrary.org/obo/cmo/releases/2019-02-19/cmo.owl>)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x rdfs:subClassOf ?y .
  OPTIONAL { ?x rdfs:label ?label }
  FILTER regex(?label, "Pituicyte count", "i")
} ORDER BY ?x
  
```

x	y
pituicyte count	pituitary gland morphological measurement
pituicyte count	pituicyte measurement
lactotroph count to total pituicyte count ratio	pituicyte count
lactotroph count to total pituicyte count ratio	calculated pituitary gland morphological measurement

[cmo](#) (<http://purl.obolibrary.org/obo/cmo/releases/2019-02-19/cmo.owl>)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x rdfs:subClassOf ?y .
  OPTIONAL { ?x rdfs:label ?label }
  FILTER regex(?label, "Benign liver tumor", "i")
} ORDER BY ?x
  
```

x	y
benign liver tumor number	liver tumor number

< > cmo (http://purl.obolibrary.org/obo/cmo/releases/2019-02-19/cmo.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x rdfs:subClassOf ?y .
  OPTIONAL { ?x rdfs:label ?label }
  FILTER regex(?label, "Egg yolk", "i")
} ORDER BY ?x

```

x	y
egg yolk weight, fowl	egg yolk measurement, fowl
egg yolk weight, fowl	part_of some 'egg weight, fowl'
egg yolk height to diameter ratio, fowl	egg yolk measurement, fowl
egg yolk measurement, fowl	egg measurement, fowl

< > cmo (http://purl.obolibrary.org/obo/cmo/releases/2019-02-19/cmo.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x rdfs:subClassOf ?y .
  OPTIONAL { ?x rdfs:label ?label }
  FILTER regex(?label, "Indirect plasma bilirubin", "i")
} ORDER BY ?x

```

x	y
indirect plasma bilirubin level	part_of some 'total plasma bilirubin level'

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x rdfs:subClassOf ?y .
  OPTIONAL { ?x rdfs:label ?label }
  FILTER regex( ?label , "Blood vessel di", "" )
} ORDER BY ?x

```

x	y
blood vessel diameter	blood vessel morphological measurement
blood vessel dilation measurement	blood vessel measurement
blood vessel distensibility	part_of some 'blood vessel resistance measurement'

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?subject ?label
WHERE {
  ?subject a owl:Class .
  OPTIONAL { ?subject rdfs:label ?label } .
  FILTER regex( ?label , "Fractio", "" )
} ORDER BY ?subject

```

subject	label
ejection fraction	"ejection fraction^^<http://www.w3.org/2001/XMLSchema#string>
fractional change in blood vessel diameter per unit change in intravascular pressure	"fractional change in blood vessel diameter per unit change in intravascular pressure^^<http://www.w3.org/2001/XMLSchema#string>

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?subject ?label
WHERE {
  ?subject a owl:Class .
  OPTIONAL { ?subject rdfs:label ?label } .
  FILTER regex( ?label , "Calculated blood vessel", "" )
} ORDER BY ?subject

```

subject	label
calculated blood vessel endothelium measurement	"calculated blood vessel endothelium measurement"
calculated blood vessel distensibility	"calculated blood vessel distensibility"

DIDEO testi ar Protege

dideo (http://purl.obolibrary.org/obo/dideo/release/2022-06-14/dideo.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?subject ?label
WHERE { ?subject a owl:Class .
OPTIONAL { ?subject rdfs:label ?label } .
FILTER regex(?label, "bioavailab", "i")
} ORDER BY ?subject
```

subject label

dideo (http://purl.obolibrary.org/obo/dideo/release/2022-06-14/dideo.owl)

> derives into

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?label
WHERE {
?x rdf:type owl:ObjectProperty
OPTIONAL { ?x rdfs:label ?label }
FILTER regex ( ?label , "orga", "i" )
}

}

x label
```

< > dideo (http://purl.obolibrary.org/obo/dideo/release/2022-06-14/dideo.owl)

> derives into

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?label
WHERE {
?x rdf:type owl:ObjectProperty
OPTIONAL { ?x rdfs:label ?label }
FILTER regex ( ?label , "cata", "i" )
}

```

x label

< > dideo (http://purl.obolibrary.org/obo/dideo/release/2022-06-14/dideo.owl)

> derives into

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?label
WHERE {
?x rdf:type owl:ObjectProperty
OPTIONAL { ?x rdfs:label ?label }
FILTER regex ( ?label , "inhib", "i" )
}

```

x label

< > dideo (http://purl.obolibrary.org/obo/dideo/release/2022-06-14/dideo.owl)

> derives into

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
?x rdfs:subClassOf ?y .
OPTIONAL { ?x rdfs:label ?label }
FILTER regex( ?label , "abs", "i" )
} ORDER BY ?x

```

x	y	label
ECO_0000267	ECO_0005517	"enzyme-linked immunoab:
ECO_0005581	ECO_0000269	"enzyme-linked immunoab:
ECO_0005581	ECO_0000267	"enzyme-linked immunoab:
ECO_0007488	ECO_0007819	"enzyme-linked immunoab:
ECO_0007488	ECO_0000267	"enzyme-linked immunoab:

dideo (http://purl.obolibrary.org/obo/dideo/release/2022-06-14/dideo.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x rdfs:subClassOf ?y .
  OPTIONAL { ?x rdfs:label ?label }
  FILTER regex( ?label , "nifed", "i" )
} ORDER BY ?x

```

x	y	label
---	---	-------

dideo (http://purl.obolibrary.org/obo/dideo/release/2022-06-14/dideo.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x rdfs:subClassOf ?y .
  OPTIONAL { ?x rdfs:label ?label }
  FILTER regex( ?label , "bioav", "i" )
} ORDER BY ?x

```

x	y
---	---

dideo (http://purl.obolibrary.org/obo/dideo/release/2022-06-14/dideo.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x rdfs:subClassOf ?y .
  OPTIONAL { ?x rdfs:label ?label }
  FILTER regex( ?label , "primary", "i" )
} ORDER BY ?x

```

x	y
GO_0044238	metabolic process
GO_0080090	● regulates some GO_0044238
GO_0080090	GO_0019222
OBL_0000811	SO_0000001
OBL_0000845	SO_0000001

dideo (http://purl.obolibrary.org/obo/dideo/release/2022-06-14/dideo.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x rdfs:subClassOf ?y .
  OPTIONAL { ?x rdfs:label ?label }
  FILTER regex(?label, "drug-drug", "i")
} ORDER BY ?x

```

x y

dideo (http://purl.obolibrary.org/obo/dideo/release/2022-06-14/dideo.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x owl:disjointWith ?y .
  OPTIONAL { ?x rdfs:label ?label }
} ORDER BY ?x

```

x	y
continuant	occurrent
occurrent	● 'part of' some continuant
independent continuant	generically dependent continuant
independent continuant	specifically dependent continuant
spatial region	site
spatial region	continuant fiat boundary
disposition	role
realizable entity	quality
specifically dependent continuant	generically dependent continuant
material entity	immaterial entity
ECO_0000000	ECO_0000217
ECO_0000033	ECO_0000034
ECO_0000203	ECO_0000218
biological_process	planned process
GO_0022411	GO_0022607
assay	OBI_0200000
OBI_0000678	OBI_0000681
ORP_0000873	ORP_0000902

Execute

dideo (http://purl.obolibrary.org/obo/dideo/release/2022-06-14/dideo.owl)

> derives into

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT *
WHERE {
  ?x rdf:type ?y .
  ?y rdf:type ?Class .
  optional { ?x rdfs:label ?lab }
  optional { ?y rdfs:label ?labl }
}

```

x	y	Class
ECO_9000000	ECO_0000217	information content entity
negatively regulates	planned process	independent continuant
RO_0002213	independent continuant	planned process
is_specified_output_of	generically dependent continuant	specifically dependent continuant
derives from	realizable entity	process
has_specified_input	function	independent continuant
concretizes	objective specification	planned process
has part	occurent	planned process
realizes		occurent
regulates		
role of		
characteristic of		
has function		
OBI_0000417		
part of		
has_specified_output		
process is result of		

Execute

[dideo](#) (<http://purl.obolibrary.org/obo/dideo/release/2022-06-14/dideo.owl>)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?range ?domain
WHERE {
?x rdf:type owl:ObjectProperty
optional { ?x rdfs:range ?range }
optional { ?x rdfs:domain ?domain }
}
  
```

x	range
concretizes	generally dependent continuant
has part	
realizes	realizable entity
regulates	
role of	
characteristic of	
has function	function
OBI_0000417	objective specification
part of	
has_specified_output	
process is result of	occurrent
participates in	occurrent
has role	role
has participant	continuant
is concretized as	specifically dependent continuant
has characteristic	specifically dependent continuant
realized in	process
function of	

Execute

< > dideo (http://purl.obolibrary.org/obo/dideo/release/2022-06-14/dideo.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT DISTINCT *
WHERE {
  ?ind a owl:NamedIndividual
  optional { ?ind rdf:type ?type }
  optional { ?type rdfs:label ?lab }
}

```

ind	owl:NamedIndividual
term split	owl:NamedIndividual
terms merged	owl:NamedIndividual
term imported	owl:NamedIndividual
placeholder removed	owl:NamedIndividual
failed exploratory term	owl:NamedIndividual
ready for release	owl:NamedIndividual
metadata incomplete	owl:NamedIndividual
metadata complete	owl:NamedIndividual
organizational term	owl:NamedIndividual
uncurated	owl:NamedIndividual
pending final vetting	owl:NamedIndividual
UO_0000019	owl:NamedIndividual
requires discussion	owl:NamedIndividual
to be replaced with external ontology term	owl:NamedIndividual
example to be eventually removed	owl:NamedIndividual
universal	owl:NamedIndividual
named class expression	owl:NamedIndividual
defined class	owl:NamedIndividual

Execute

< > dideo (http://purl.obolibrary.org/obo/dideo/release/2022-06-14/dideo.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?subject ?label
WHERE { ?subject a owl:Class .
OPTIONAL { ?subject rdfs:label ?label } .
FILTER regex(?label , "cryopre", "i" )
} ORDER BY ?subject

```

subject	label
---------	-------

< > dideo (http://purl.obolibrary.org/obo/dideo/release/2022-06-14/dideo.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?subject ?label
WHERE { ?subject a owl:Class .
OPTIONAL { ?subject rdfs:label ?label } .
FILTER regex(?label, "human", "i")
} ORDER BY ?subject

```

subject	label
---------	-------

< > dideo (http://purl.obolibrary.org/obo/dideo/release/2022-06-14/dideo.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?subject ?label
WHERE { ?subject a owl:Class
OPTIONAL { ?subject rdfs:label ?label }
FILTER regex(?label, "cryopre", "i")
}

```

subject	label
cryopreserved human hepatocyte population	"cryopreserved human hepatocyte population"@en

dideo (http://purl.obolibrary.org/obo/dideo/release/2022-06-14/dideo.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?subject ?object ?label

WHERE { ?subject rdfs:subClassOf ?object
OPTIONAL {?subject rdfs:label ?label}
FILTER regex(?label, "abs", "i")
}

```

subject	object
abstract	document part
abstract	● 'is about' some document
drug absorbability assay	assay
absorbability measurement datum	● is about' some 'drug product
drug absorbability assay	● has_specified_input some (absorbability and ('cha
absorbability	disposition
absorbability measurement datum	scalar measurement datum
absorption of material	process

DOID testi ar Protege

doid (http://purl.obolibrary.org/obo/doid/releases/2022-09-29/doid.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?subject ?label
WHERE { ?subject a owl:Class .
OPTIONAL { ?subject rdfs:label ?label } .
FILTER regex( ?label , "3-m", "i" )
} ORDER BY ?subject

```

subject	label
---------	-------

[doid \(http://purl.obolibrary.org/obo/doid/releases/2022-09-29/doid.owl\)](#)

[Active ontology](#) x [Entities](#) x [Individuals by class](#) x [DL Query](#) x [SPARQL Query](#) x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?subject ?label
WHERE { ?subject a owl:Class .
OPTIONAL { ?subject rdfs:label ?label } .
FILTER regex( ?label , "acer", "i" )
} ORDER BY ?subject
  
```

subject	label
---------	-------

[doid \(http://purl.obolibrary.org/obo/doid/releases/2022-09-29/doid.owl\)](#)

[Active ontology](#) x [Entities](#) x [Individuals by class](#) x [DL Query](#) x [SPARQL Query](#) x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?subject ?label
WHERE { ?subject a owl:Class .
OPTIONAL { ?subject rdfs:label ?label } .
FILTER regex( ?label , "acheiro", "i" )
} ORDER BY ?subject
  
```

subject	label
---------	-------

[doid \(http://purl.obolibrary.org/obo/doid/releases/2022-09-29/doid.owl\)](#)

[Active ontology](#) x [Entities](#) x [Individuals by class](#) x [DL Query](#) x [SPARQL Query](#) x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?label
WHERE {
?x rdf:type owl:ObjectProperty
OPTIONAL { ?x rdfs:label ?label } .
FILTER regex( ?label , "disease relat", "i" )
}
  
```

x	label
---	-------

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

```
SELECT distinct ?x ?label
WHERE {
  ?x rdf:type owl:ObjectProperty
  OPTIONAL { ?x rdfs:label ?label } .
  FILTER regex( ?label , "derives from", ";" )
}
```

x
derives from

label
"derives from"@en

< > [doid \(http://purl.obolibrary.org/obo/doid/releases/2022-09-29/doid.owl\)](http://purl.obolibrary.org/obo/doid/releases/2022-09-29/doid.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

```
SELECT distinct ?x ?label
WHERE {
  ?x rdf:type owl:ObjectProperty
  OPTIONAL { ?x rdfs:label ?label } .
  FILTER regex( ?label , "condition", "i" )
}
```

x

label

< > doid (http://purl.obolibrary.org/obo/doid/releases/2022-09-29/doid.o

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x rdfs:subClassOf ?y .
  OPTIONAL { ?x rdfs:label ?label }
  FILTER regex( ?label , "Achroma", "i" )
} ORDER BY ?x

```

x y

< > doid (http://purl.obolibrary.org/obo/doid/releases/2022-09-29/doid.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x rdfs:subClassOf ?y .
  OPTIONAL { ?x rdfs:label ?label }
  FILTER regex( ?label , "absolu", "i" )
} ORDER BY ?x

```

x y

< > doid (http://purl.obolibrary.org/obo/doid/releases/2022-09-29/doid.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
?x rdfs:subClassOf ?y .
OPTIONAL { ?x rdfs:label ?label }
FILTER regex( ?label , "primary cong", "i" )
} ORDER BY ?x

```

x y

< > doid (http://purl.obolibrary.org/obo/doid/releases/2022-09-29/doid.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
?x rdfs:subClassOf ?y .
OPTIONAL { ?x rdfs:label ?label }
FILTER regex( ?label , "hypocho", "i" )
} ORDER BY ?x

```

x y

< > doid (http://purl.obolibrary.org/obo/doid/releases/2022-09-29/doid.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
?x rdfs:subClassOf ?y .
OPTIONAL { ?x rdfs:label ?label }
FILTER regex(?label, "pvcn", "i" )
} ORDER BY ?x

```

x y

< > doid (http://purl.obolibrary.org/obo/doid/releases/2022-09-29/doid.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
?x owl:disjointWith ?y .
OPTIONAL { ?x rdfs:label ?label }
} ORDER BY ?x

```

x	y	label
BFO_0000002	● 'part of some BFO_0000003	"continuant"@en
BFO_0000002	BFO_0000003	"continuant"@en
BFO_0000003	● 'part of some BFO_0000002	"occurent"@en
BFO_0000004	BFO_0000031	"independent continuant"@en
BFO_0000004	BFO_0000020	"independent continuant"@en
BFO_0000006	BFO_0000029	"spatial region"@en
BFO_0000006	BFO_0000140	"spatial region"@en
BFO_0000016	BFO_0000023	"disposition"@en
BFO_0000017	BFO_0000019	"realizable entity"@en
BFO_0000020	BFO_0000031	"specifically dependent continuant"@en
BFO_0000040	BFO_0000141	"material entity"@en
evidence	ECO_0000217	"evidence"^^<http://www.w3.org/2001/
ECO_0000033	ECO_0000034	"author statement supported by traceable
ECO_0000203	ECO_0000218	"automatic assertion"^^<http://www.w3.c
GO_0008150	OBI_0000011	"biological_process"
GO_0022411	GO_0022607	"cellular component disassembly"
OBI_0000070	OBI_0200000	"assay"@en
OBI_0000678	OBI_0000681	"nontoxic substance"

eco (http://purl.obolibrary.org/obo/eco/releases/2022-10-10/eco.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?subject
WHERE { ?subject a owl:Class .
OPTIONAL { ?subject rdfs:label ?label } .
FILTER regex( ?label , "4C evidence", "i" )
} ORDER BY ?subject
```

subject
4C evidence

eco (http://purl.obolibrary.org/obo/eco/releases/2022-10-10/eco.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?subject
WHERE { ?subject a owl:Class .
OPTIONAL { ?subject rdfs:label ?label } .
FILTER regex( ?label , "3C evidence", "i" )
} ORDER BY ?subject
```

subject
3C evidence

eco (http://purl.obolibrary.org/obo/eco/releases/2022-10-10/eco.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?subject
WHERE { ?subject a owl:Class .
OPTIONAL { ?subject rdfs:label ?label }.
FILTER regex( ?label , "DNA aff", "i" )
} ORDER BY ?subject

```

subject
DNA affinity chromatography evidence

eco (http://purl.obolibrary.org/obo/eco/releases/2022-10-10/eco.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x
WHERE {
?x rdf:type owl:ObjectProperty }

```

x
has quantity
located in
bound_to
preceded by
immediately preceded by
has measurement unit label
is quality measurement of
has grain
temporal relation
quality of
objective_achieved_by
immediately precedes
precedes
is_supported_by_data
is quality measured as
member of
has member
uses

Execute

eco (http://purl.obolibrary.org/obo/eco/releases/2022-10-10/eco.owl)

Active ontology x | Entities x | Individuals by class x | DL Query x | SPARQL Query x

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x
WHERE {
?x rdf:type owl:ObjectProperty }
```

x

used_in

eco (http://purl.obolibrary.org/obo/eco/releases/2022-10-10/eco.owl)

Active ontology x | Entities x | Individuals by class x | DL Query x | SPARQL Query x

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
?x rdfs:subClassOf ?y .
OPTIONAL { ?x rdfs:label ?label }
FILTER regex(?label , "clinical study evidence used in auto", "i")
} ORDER BY ?x
```

x	y
clinical study evidence used in automatic assertion	experimental evidence used in automatic assertion
clinical study evidence used in automatic assertion	clinical study evidence

eco (http://purl.obolibrary.org/obo/eco/releases/2022-10-10/eco.owl)

Active ontology x | Entities x | Individuals by class x | DL Query x | SPARQL Query x

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
?x rdfs:subClassOf ?y .
OPTIONAL { ?x rdfs:label ?label }
FILTER regex(?label , "polysome profiling evidence used in automatic ass", "i")
} ORDER BY ?x
```

x	y
polysome profiling evidence used in automatic assertion	experimental evidence used in automatic assertion
polysome profiling evidence used in automatic assertion	polysome profiling evidence

eco (http://purl.obolibrary.org/obo/eco/releases/2022-10-10/eco.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
?x rdfs:subClassOf ?y .
OPTIONAL { ?x rdfs:label ?label }
FILTER regex( ?label , "quantitative macroscopy evidence used in man", "i" )
} ORDER BY ?x

```

x	y
quantitative macroscopy evidence used in manual assertion	quantitative macroscopy evidence
quantitative macroscopy evidence used in manual assertion	macroscopy evidence used in manual assertion

eco (http://purl.obolibrary.org/obo/eco/releases/2022-10-10/eco.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
?x rdfs:subClassOf ?y .
OPTIONAL { ?x rdfs:label ?label }
FILTER regex( ?label , "amplification", "i" )
} ORDER BY ?x

```

x	y
amplification of intermethylated sites evidence	● is_specified_output_of some (assay and (re
amplification of intermethylated sites evidence	● is about' some 'DNA methylation

eco (http://purl.obolibrary.org/obo/eco/releases/2022-10-10/eco.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
?x rdfs:subClassOf ?y .
OPTIONAL { ?x rdfs:label ?label }
FILTER regex(?label, "transcriptional", "i")
} ORDER BY ?x

```

x	y
transcriptional activation assay evidence	is_specified_output_of some (assay and (realizes some (concretizes some 'tr
transcriptional activation assay evidence	is about' some 'core promoter sequence-specific DNA binding

eco (http://purl.obolibrary.org/obo/eco/releases/2022-10-10/eco.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
?x rdfs:subClassOf ?y .
OPTIONAL { ?x rdfs:label ?label }
FILTER regex(?label, "cell transfection", "i")
} ORDER BY ?x

```

x	y
cell transfection experiment evidence	is_specified_output_of some ('in live cell assay' and ('has participant' some 'g
cell transfection experiment evidence	is_specified_output_of some (assay and ('has part' some transfection))
cell transfection experiment evidence	is_specified_output_of some (assay and (realizes some (concretizes some 'ge
cell transfection experiment evidence	is about' some 'regulation of biological process

eco (http://purl.obolibrary.org/obo/eco/releases/2022-10-10/eco.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x owl:disjointWith ?y .
  OPTIONAL { ?x rdfs:label ?label }
  FILTER regex(?label, "", "i" )
} ORDER BY ?x

```

x	y
continuant	part of some occurrent
continuant	occurrent
occurent	part of some continuant
independent continuant	generically dependent continuant
independent continuant	specifically dependent continuant
spatial region	site
spatial region	continuant flat boundary
disposition	role
realizable entity	quality
specifically dependent continuant	generically dependent continuant
material entity	immaterial entity
evidence	assertion method
author statement supported by traceable reference	author statement without traceable support
automatic assertion	manual assertion

Entity	Id	Match
Display name: universal	IAO:0000410	universal
definition: defined class	IAO:0000420	definition [language: en] A defined class is a class tha
rdfs:comment: rapid amplification of cDNA ends polymerase chain reaction evidence	ECO:0001820	rdfs:comment [type: xsd:string] 5' and 3' RACE-PCR
rdfs:label: universal	IAO:0000410	rdfs:label [language: en] universal
EquivalentClasses: denotator type	IAO:0000409	'denotator type' EquivalentTo (universal, 'defined clas

eco (http://purl.obolibrary.org/obo/eco/releases/2022-10-10/eco.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT *
WHERE {
  ?x rdf:type ?y .
  ?y rdf:type ?Class .
  optional { ?x rdfs:label ?lab }
  optional { ?y rdfs:label ?labl }
}

```

x	y	Class	lab

placeholder removed

Case sensitive
 Whole words
 Ignore white space
 Regular expression
 Show all results

Search in IRIs
 Search in annotation values
 Search in logical axioms

Found in	Entity	Id	Match
Display name	placeholder removed	IAO:0000226	placeholder removed
rdfs:label	placeholder removed	IAO:0000226	rdfs:label [language: en] placeholder
EquivalentClasses	obsolescence reason specification	IAO:0000225	'obsolescence reason specification'

uncurated

Case sensitive
 Whole words
 Ignore white space
 Regular expression
 Show all results

Search in IRIs
 Search in annotation values
 Search in logical axioms

Found in	Entity	Id	Match
Display name	uncurated	IAO:0000124	uncurated
rdfs:label	uncurated	IAO:0000124	rdfs:label [language: en] uncurated
EquivalentClasses	'curation status specification'	IAO:0000078	'curation status specification' Equiva

[<](#)
[>](#)
[eco \(http://purl.obolibrary.org/obo/eco/releases/2022-10-10/eco.owl\)](http://purl.obolibrary.org/obo/eco/releases/2022-10-10/eco.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?range ?domain
WHERE {
  ?x rdf:type owl:ObjectProperty .
  ?x rdfs:domain ?domain
}

```

x	range	domain
used_in		evidence
achieves_planned_objective		planned process
has participant		occurent
participates in		continuant
function of		function
located in		'independent continuan
preceded by		occurent
immediately preceded by		occurent

[<](#)
[>](#)
[eco \(http://purl.obolibrary.org/obo/eco/releases/2022-10-10/eco.owl\)](http://purl.obolibrary.org/obo/eco/releases/2022-10-10/eco.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?range ?domain
WHERE {
  ?x rdf:type owl:ObjectProperty
  optional { ?x rdfs:range ?range }
  optional { ?x rdfs:domain ?domain }
}

```

x	range
has role	role
has participant	continuant
is concretized as	specifically dependent continuant
bearer of	specifically dependent continuant
realized in	process
function of	
location of	
has value specification	value specification
is_specified_input_of	planned process
derives into	
has quality	quality
located in	'independent continuant' and (not ('spatial region'))
bound_to	material entity
preceded by	occurent
immediately preceded by	
has measurement unit label	measurement unit label
is quality measurement of	quality
has origin	

Execute

GENO testi ar Protege

geno (http://purl.obolibrary.org/obo/geno/releases/2022-08-10/geno.owl)

Active ontology x | Entities x | Individuals by class x | DL Query x | SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x owl:disjointWith ?y .
  OPTIONAL { ?x rdfs:label ?label }
  FILTER regex (?label, "contin", "i")
} ORDER BY ?x
    
```

x	y	label
continuant	occurent	"continuant"@en
continuant	occurent	"continuant"
continuant	● is part of some occurent	"continuant"@en
continuant	● is part of some occurent	"continuant"
independent continuant	generically dependent continuant	"independent continuant"
independent continuant	generically dependent continuant	"independent continuant"@en
independent continuant	specifically dependent continuant	"independent continuant"
independent continuant	specifically dependent continuant	"independent continuant"@en
specifically dependent continuant	generically dependent continuant	"specifically dependent continuant"
specifically dependent continuant	generically dependent continuant	"specifically dependent continuant"@en

geno (http://purl.obolibrary.org/obo/geno/releases/2022-08-10/geno.owl)

Active ontology x | Entities x | Individuals by class x | DL Query x | SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x owl:disjointWith ?y .
  OPTIONAL { ?x rdfs:label ?label }
  FILTER regex (?label, "quality", "i")
} ORDER BY ?x
    
```

x	y	label

geno (http://purl.obolibrary.org/obo/geno/releases/2022-08-10/geno.owl)

Active ontology x | Entities x | Individuals by class x | DL Query x | SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x owl:disjointWith ?y .
  OPTIONAL { ?x rdfs:label ?label }
  FILTER regex (?label, "sequence feature", "i")
} ORDER BY ?x
    
```

x	y	label
sequence feature location	sequence_feature	"sequence feature location"@en

geno (http://purl.obolibrary.org/obo/geno/releases/2022-08-10/geno.owl)

Individuals by class | DL Query | SPARQL Query

Datatypes | Individuals

Search

exploratory term

Case sensitive Whole words Ignore white space Regular expression Show all results

Search in IRIs Search in annotation values Search in logical axioms

Found in	Entity	Id	Match
	exploratory term	GENO:0000484	exploratory term
	failed exploratory term	IAO:0000103	failed exploratory term
'editor preferred term'	failed exploratory term	IAO:0000103	'editor preferred term' [language: en]
rdfs:label	exploratory term	GENO:0000484	rdfs:label [language: en] exploratory term
	failed exploratory term	IAO:0000103	rdfs:label [language: en] failed exploratory term
ClassAssertion	exploratory term	GENO:0000484	'exploratory term' Type 'data about'
	failed exploratory term	IAO:0000103	'failed exploratory term' Type 'obsolete'
EquivalentClasses	obsolescence reason specification	IAO:0000225	'obsolescence reason specification'

geno (http://purl.obolibrary.org/obo/geno/releases/2022-08-10/geno.owl)

Individuals by class | DL Query | SPARQL Query

Datatypes | Individuals

Search

universal

Case sensitive Whole words Ignore white space Regular expression Show all results

Search in IRIs Search in annotation values Search in logical axioms

Found in	Entity	Id	Match
	universal	IAO:0000410	universal
'editor note'	defined class	IAO:0000420	'editor note' [language: en] 'defined class'
	entity	BFO:0000001	'editor note' [language: en] BFO:0000001
	role	BFO:0000023	'editor note' [language: en] BFO:0000023
'editor preferred term'	universal	IAO:0000410	'editor preferred term' [language: en]
definition	defined class	IAO:0000420	definition [language: en] A defined class
rdfs:label	universal	IAO:0000410	rdfs:label [language: en] universal
ClassAssertion	universal	IAO:0000410	universal Type 'denominator type'
EquivalentClasses	denominator type	IAO:0000409	'denominator type' EquivalentTo (universal)

geno (http://purl.obolibrary.org/obo/geno/releases/2022-08-10/geno.owl)

Individuals by class | DL Query | SPARQL Query

Datatypes | Individuals

Search

defined class type

Case sensitive Whole words Ignore white space Regular expression Show all results

Search in IRIs Search in annotation values Search in logical axioms

Found in	Entity	Id	Match
'editor note'	haplotype	GENO:0000871	'editor note' Consider if we don't
	'lost aneusomic chromosome'	GENO:0000339	'editor note' This 'lost' chromoso
	'variant copy number complement'	GENO:0000962	'editor note' Note that this 'varia
	'environmental system'	ENVO:01000254	'editor note' 1. Stub class to serv
	is_allele_of	GENO:0000408	'editor note' Note that the allele
	'haplotype block'	GENO:0000898	'editor note' Consider whether w
definition	'is denominator type'	IAO:0000411	definition [language: en] Relates
rdfs:comment	obsolete haplotype block	GENO:0000916	rdfs:comment A haplotype block
	genotype	GENO:0000536	rdfs:comment 1. Scope of 'Geneti
	sequence_alteration	SO:0001059	rdfs:comment 1. A 'sequence alte
	is_allele_of	GENO:0000408	rdfs:comment To allow users to r
ClassAssertion	defined class	IAO:0000420	'defined class' Type 'denominator ty
EquivalentClasses	denominator type	IAO:0000409	'denominator type' EquivalentTo (un

geno (http://purl.obolibrary.org/obo/geno/releases/2022-08-10/geno.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?range ?domain
WHERE {
  ?x rdf:type owl:ObjectProperty
  optional { ?x rdfs:range ?range }
  optional { ?x rdfs:domain ?domain }
}

```

x	range
is_reference_allele_of	
has_asserted_phenotype	
has_sequence_unit	
has_defining_location	sequence feature location
is_allele_of	
has_sequence	biological sequence

geno (http://purl.obolibrary.org/obo/geno/releases/2022-08-10/geno.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?range ?domain
WHERE {
  ?x rdf:type owl:ObjectProperty
  optional { ?x rdfs:range ?range }
  optional { ?x rdfs:domain ?domain }
}

```

x	range	domain
has_qualifying_environment		
is_genotype_of		
is_polymorphic_allele_of		polymorphic allele
is downstream of sequence of		
is upstream of sequence of		
causes or contributes to condition		
has subsequence		
molecularly controls		
obsolete_is_variant_with		
has_interval		sequence feature location
has_reference_sequence		

geno (http://purl.obolibrary.org/obo/geno/releases/2022-08-10/geno.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?subject ?label
WHERE { ?subject a owl:Class .
OPTIONAL { ?subject rdfs:label ?label } .
FILTER regex( ?label , "", "i" )
} ORDER BY ?subject

```

subject	label
biological sequence	"biological sequence"@en
obsolete biological sequence or collection	"obsolete biological sequence or collection"^^<http://www.w3.org/2001/XMLSchema#string>
obsolete biological sequence collection	"obsolete biological sequence collection"^^<http://www.w3.org/2001/XMLSchema#string>
biological sequence unit	"biological sequence unit"@en
biological sequence or set	"biological sequence or set"@en
biological sequence set	"biological sequence set"@en

geno (http://purl.obolibrary.org/obo/geno/releases/2022-08-10/geno.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?subject ?label
WHERE { ?subject a owl:Class .
OPTIONAL { ?subject rdfs:label ?label } .
FILTER regex( ?label , "obsolete genomic feature", "i" )
} ORDER BY ?subject

```

subject	label
obsolete genomic feature collection	"obsolete genomic feature collection"^^<http://www.w3.org/2001/XMLSchema#string>

geno (http://purl.obolibrary.org/obo/geno/releases/2022-08-10/geno.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?label
WHERE {
?x rdf:type owl:ObjectProperty
OPTIONAL { ?x rdfs:label ?label } .
FILTER regex( ?label , "denotes", "i" )
}

```

x	label
denotes	"denotes"

geno (http://purl.obolibrary.org/obo/geno/releases/2022-08-10/geno.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?label
WHERE {
?x rdf:type owl:ObjectProperty
OPTIONAL { ?x rdfs:label ?label } .
FILTER regex( ?label , "expand", "i" )
}
```

x	label

geno (http://purl.obolibrary.org/obo/geno/releases/2022-08-10/geno.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
?x rdfs:subClassOf ?y .
OPTIONAL { ?x rdfs:label ?label }
FILTER regex( ?label , "diplotype", "i" )
} ORDER BY ?x
```

x	y	label
diplotype	● denotes some 'single locus complement' and ('has member' exactly 2 allele);	"diplotype"@gen
diplotype	allelic genotype	"diplotype"@gen

geno (http://purl.obolibrary.org/obo/geno/releases/2022-08-10/geno.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
?x rdfs:subClassOf ?y .
OPTIONAL { ?x rdfs:label ?label }
FILTER regex( ?label , "allelic", "i" )
} ORDER BY ?x
```

x	y	label
allelic genotype	intrinsic genotype	"allelic genotype"@en
allelic genotype	● denotes some 'single locus complement'	"allelic genotype"@en
allelic genotype	genomic entity	"allelic genotype"@en
allelic state	sequence feature attribute	"allelic state"@en
allelic phase	sequence feature attribute	"allelic phase"
allelic cellular distribution	sequence feature attribute	"allelic cellular distribution"@en

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x rdfs:subClassOf ?y .
  OPTIONAL { ?x rdfs:label ?label }
  FILTER regex( ?label , "genomic background", "i" )
} ORDER BY ?x

```

x	y	label
genomic background	● 'sequence role' some reference	"genomic background"@en
genomic background	genomic genotype	"genomic background"@en
genomic background	● denotes some 'background genome'	"genomic background"@en
unspecified genomic background	genomic background	"unspecified genomic background"@en

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x rdfs:subClassOf ?y .
  OPTIONAL { ?x rdfs:label ?label }
  FILTER regex( ?label , "allelic gen", "i" )
} ORDER BY ?x

```

x	y	label
allelic genotype	intrinsic genotype	"allelic genotype"@en
allelic genotype	● denotes some 'single locus complement'	"allelic genotype"@en
allelic genotype	genomic entity	"allelic genotype"@en

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x rdfs:subClassOf ?y .
  OPTIONAL { ?x rdfs:label ?label }
  FILTER regex( ?label , "variant allele", "i" )
} ORDER BY ?x

```

x	y	label
variant allele	● varies_with some 'reference allele'	"variant allele"
variant allele	allele	"variant allele"
variant allele	● sequence_alteration or ('has subsequence' some sequence_alteration)	"variant allele"

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x rdfs:subClassOf ?y .
  OPTIONAL { ?x rdfs:label ?label }
  FILTER regex( ?label , "amino acid", "i" )
} ORDER BY ?x

```

x	y	label
ECO_0001109	ECO_0000002	"phosphoamino acid analysis evidence"^^<http://www.w3.org/2001/XMLSchema#textLiteral>
ECO_0001243	ECO_0000314	"phosphoamino acid analysis evidence used in manual assertion"^^<http://www.w3.org/2001/XMLSchema#textLiteral>
ECO_0001243	ECO_0001109	"phosphoamino acid analysis evidence used in manual assertion"^^<http://www.w3.org/2001/XMLSchema#textLiteral>
ECO_0007444	ECO_0007307	"phosphoamino acid analysis evidence used in automatic assertion"^^<http://www.w3.org/2001/XMLSchema#textLiteral>
ECO_0007444	ECO_0001109	"phosphoamino acid analysis evidence used in automatic assertion"^^<http://www.w3.org/2001/XMLSchema#textLiteral>
amino acid sequence	biological sequence	"amino acid sequence"@en
amino acid sequence	has_sequence_unit some "amino acid residue"	"amino acid sequence"@en
amino acid residue	biological sequence unit	"amino acid residue"@en

HSO testi ar Protege

hso (http://purl.obolibrary.org/obo/hso/2021-12-13/hso.owl)

> superficial_to

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT DISTINCT *
WHERE {
  ?ind a owl:NamedIndividual
  optional { ?ind rdf:type ?type }
  optional { ?ind rdfs:label ?lab }
  FILTER regex( ?lab , "Brazil", "i" )
}

```

ind	type
Brazil	Country
Brazil	owl:NamedIndividual

hso (http://purl.obolibrary.org/obo/hso/2021-12-13/hso.owl)

superficial_to

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT DISTINCT *
WHERE {
?ind a owl:NamedIndividual
optional { ?ind rdf:type ?type }
optional { ?ind rdfs:label ?lab }
FILTER regex( ?lab , "China", "i" )
}

```

ind	type
China	Country
China	owl:NamedIndividual
Republic of China	owl:NamedIndividual
Republic of China	Country

hso (http://purl.obolibrary.org/obo/hso/2021-12-13/hso.owl)

superficial_to

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?range ?domain
WHERE {
?x rdf:type owl:ObjectProperty
optional { ?x rdfs:range ?range }
optional { ?x rdfs:domain ?domain }
}

```

x	range	domain
function of		function
has participant	continuant	occurent
transformation of		
ends with		
place of exposure	organization	Surveillance Activity
has surveillance context	surveillance context	Surveillance Activity' or 'surveillance protocol
bearer of	specifically dependent continuant	
target pathogen	organism	Surveillance Activity
continues with		

hso (http://purl.obolibrary.org/obo/hso/2021-12-13/hso.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?label
WHERE {
  ?x rdf:type owl:Class
  OPTIONAL { ?x rdfs:label ?label } .
  FILTER regex( ?label , "contin", "i" )
}

```

x	label
generically dependent continuant	"generically dependent continuant"@en
continuant	"continuant"@en
independent continuant	"independent continuant"@en
continuant fiat boundary	"continuant fiat boundary"@en
specifically dependent continuant	"specifically dependent continuant"@en

hso (http://purl.obolibrary.org/obo/hso/2021-12-13/hso.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x rdfs:subClassOf ?y .
  OPTIONAL { ?x rdfs:label ?label }
  FILTER regex( ?label , "Spatial", "i" )
} ORDER BY ?x

```

x	y
spatial region	immaterial entity

hso (http://purl.obolibrary.org/obo/hso/2021-12-13/hso.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
?x owl:disjointWith ?y .
OPTIONAL { ?x rdfs:label ?label }
FILTER regex ( ?label , "Dispo", "i" )
} ORDER BY ?x

```

x	y
disposition	role

hso (http://purl.obolibrary.org/obo/hso/2021-12-13/hso.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
?x owl:disjointWith ?y .
OPTIONAL { ?x rdfs:label ?label }
FILTER regex ( ?label , "Census", "i" )
} ORDER BY ?x

```

x	y
---	---

hso (http://purl.obolibrary.org/obo/hso/2021-12-13/hso.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?range
WHERE {
  ?x rdf:type owl:ObjectProperty .
  ?x rdfs:range ?range
}

```

x	range
is_specified_input_of	planned process
OBL_0001938	OBL_0001933
has role	role
RO_0000085	function
RO_0000086	quality
is_supported_by_data	data item
objective_achieved_by	planned process
realized in	process
realizes	realizable entity
IAO_0000221	quality

hso (http://purl.obolibrary.org/obo/hso/2021-12-13/hso.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?subject ?label
WHERE {
  ?subject a owl:Class .
  OPTIONAL { ?subject rdfs:label ?label } .
  FILTER regex( ?label , "Extent", "i" )
} ORDER BY ?subject

```

subject	label
Extent of outbreak	"Extent of outbreak"@en

hso (http://purl.obolibrary.org/obo/hso/2021-12-13/hso.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?subject ?label
WHERE {
  ?subject a owl:Class .
  OPTIONAL { ?subject rdfs:label ?label } .
  FILTER regex( ?label , "Fluid", "i" )
} ORDER BY ?subject

```

subject	label
fluid specimen	"fluid specimen"@en
fluid specimen	"fluid specimen"@en
haemolymphatic fluid	"haemolymphatic fluid"^^<http://www.w3.org/2001/XMLSchema#string>
interstitial fluid	"interstitial fluid"^^<http://www.w3.org/2001/XMLSchema#string>
synovial fluid	"synovial fluid"^^<http://www.w3.org/2001/XMLSchema#string>
bodily fluid	"bodily fluid"^^<http://www.w3.org/2001/XMLSchema#string>
mammary gland fluid/secretion	"mammary gland fluid/secretion"^^<http://www.w3.org/2001/XMLSchema#string>

hso (http://purl.obolibrary.org/obo/hso/2021-12-13/hso.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?label
WHERE {
?x rdf:type owl:ObjectProperty
OPTIONAL { ?x rdfs:label ?label }.
FILTER regex( ?label , "Available", "i" )
}

```

x	label
available in language	"available in language"@en

hso (http://purl.obolibrary.org/obo/hso/2021-12-13/hso.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?label
WHERE {
?x rdf:type owl:ObjectProperty
OPTIONAL { ?x rdfs:label ?label }.
FILTER regex( ?label , "Sampling", "i" )
}

```

x	label
has sampling design	"has sampling design"@en
has sampling unit	"has sampling unit"@en
applies sampling strategy	"applies sampling strategy"@en
sampling stage	"sampling stage"@en

hso (http://purl.obolibrary.org/obo/hso/2021-12-13/hso.owl)

superficial_to

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?label
WHERE {
?x rdf:type owl:ObjectProperty
OPTIONAL { ?x rdfs:label ?label }.
FILTER regex( ?label , "Superficial", "i" )
}

```

x	label
superficial_to	"superficial_to"^^<http://www.w3.org/2001/XMLSchema#string>

hso (http://purl.obolibrary.org/obo/hso/2021-12-13/hso.owl)

superficial_to

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
?x rdfs:subClassOf ?y .
OPTIONAL { ?x rdfs:label ?label }
FILTER regex( ?label , "Temporal", "i" )
} ORDER BY ?x

```

x	y
temporal region	occurrent
spatiotemporal region	occurrent
one-dimensional temporal region	temporal region
zero-dimensional temporal region	temporal region

hso (http://purl.obolibrary.org/obo/hso/2021-12-13/hso.owl)

superficial_to

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
?x rdfs:subClassOf ?y .
OPTIONAL { ?x rdfs:label ?label }
FILTER regex( ?label , "Town", "i" )
} ORDER BY ?x

```

x	y
township	dense settlement
township	Municipality

hso (http://purl.obolibrary.org/obo/hso/2021-12-13/hso.owl)

superficial_to

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x rdfs:subClassOf ?y .
  OPTIONAL { ?x rdfs:label ?label }
  FILTER regex( ?label , "Blood", "i" )
} ORDER BY ?x

```

x	y
blood	part of some 'hematopoietic system'
blood	haemolymphatic fluid
blood	has part some 'blood plasma'
blood plasma	haemolymphatic fluid
blood plasma	part of some blood
venous blood	blood

hso (http://purl.obolibrary.org/obo/hso/2021-12-13/hso.owl)

superficial_to

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x rdfs:subClassOf ?y .
  OPTIONAL { ?x rdfs:label ?label }
  FILTER regex( ?label , "Transu", "i" )
} ORDER BY ?x

```

x	y
transudate	bodily fluid
transudate	develops_from some 'blood plasma'
transudate	transformation of some 'blood plasma'

hso (http://purl.obolibrary.org/obo/hso/2021-12-13/hso.owl)

superficial_to

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
?x rdfs:subClassOf ?y .
OPTIONAL { ?x rdfs:label ?label }
FILTER regex( ?label , "Inters", "i" )
} ORDER BY ?x

```

x	y
Interstate/ interprovincial travel surveillance	Domestic travel surveillance
interstitial fluid	organism substance
interstitial fluid	'part of' some tissue

hso (http://purl.obolibrary.org/obo/hso/2021-12-13/hso.owl)

superficial_to

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
?x owl:disjointWith ?y .
OPTIONAL { ?x rdfs:label ?label }
FILTER regex ( ?label , "Assay", "i" )
} ORDER BY ?x

```

x	y
assay	planning

hso (http://purl.obolibrary.org/obo/hso/2021-12-13/hso.owl)

> superficial_to

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT DISTINCT *
WHERE {
  ?ind a owl:NamedIndividual
  optional { ?ind rdf:type ?type }
  optional { ?ind rdfs:label ?lab }
  FILTER regex( ?lab , "Belg", "i" )
}

```

ind	type
Belgium	Country
Belgium	owl:NamedIndividual

IDO testi ar Protege

ido (http://purl.obolibrary.org/obo/ido/2017-11-03/ido.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x rdfs:subClassOf ?y .
  OPTIONAL { ?x rdfs:label ?label }
  FILTER regex( ?label , "infection incide", "i" )
} ORDER BY ?x

```

x	y
infection incidence	inheres in' some 'organism population quality
infection incidence	quality
infection incidence proportion	quality
infection incidence proportion	inheres in' some 'organism population
infection incidence rate	inheres in' some 'organism population
infection incidence rate	quality

ido (http://purl.obolibrary.org/obo/ido/2017-11-03/ido.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?subject ?label
WHERE { ?subject a owl:Class .
OPTIONAL { ?subject rdfs:label ?label } .
FILTER regex( ?label , "infection incidence", "i" )
} ORDER BY ?subject

```

subject	label
---------	-------

ido (http://purl.obolibrary.org/obo/ido/2017-11-03/ido.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
?x rdfs:subClassOf ?y .
OPTIONAL { ?x rdfs:label ?label }
FILTER regex( ?label , "infect", "i" )
} ORDER BY ?x

```

x	y	label
ECO_0001833	ECO_0001830	"infection assay evidence"^^<http://www.w3.org/2001/XMLSchema#string>
ECO_0001834	ECO_0001831	"infection assay evidence used in automatic assertion"
ECO_0001834	ECO_0001833	"infection assay evidence used in manual assertion"
ECO_0001835	ECO_0001832	"infection assay evidence used in manual assertion"
ECO_0001835	ECO_0001833	"infection"@@en
infection	material entity	"infectious agent"@@en
OBL_0000925	organism	"infection process"@@en
OBL_1110021	OBL_1110122	"infection process"@@en
OBL_1110021	● realizes some OBL_1110093	"infection process"@@en
OBL_1110021	● 'has participant' some infection	"infection process"@@en
OBL_1110030	● realizes some (OBL_0000319 and ('inheres in' some OBL_0000925))	"experimental infection of cell culture"@@en
OBL_1110030	OBL_0600000	"experimental infection of cell culture"@@en
OBL_1110093	disposition	"disposition to infect an organism"@@en
OBL_1110093	● 'realized in' only OBL_1110021	"disposition to infect an organism"@@en

ido (http://purl.obolibrary.org/obo/ido/2017-11-03/ido.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x rdfs:subClassOf ?y .
  OPTIONAL { ?x rdfs:label ?label }
  FILTER regex( ?label , "source", "i" )
} ORDER BY ?x

```

x	y
source of infection role	role
source of infection role	inheres in' some 'material entity
source of infection	material entity

ido (http://purl.obolibrary.org/obo/ido/2017-11-03/ido.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x rdfs:subClassOf ?y .
  OPTIONAL { ?x rdfs:label ?label }
  FILTER regex( ?label , "cyto", "i" )
} ORDER BY ?x

```

x	y
FtsZ-dependent cytokinesis	reproduction
_obsolete_transcytosis	Obsolete Class
_obsolete_virus-induced cytopathogenesis	Obsolete Class
_obsolete_resist phagocytosis in subepithelial space	Obsolete Class
cytotoxin	toxin
cytotoxin	has disposition' some 'toxin disposition

ido (http://purl.obolibrary.org/obo/ido/2017-11-03/ido.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
?x owl:disjointWith ?y .
OPTIONAL { ?x rdfs:label ?label }
} ORDER BY ?x
```

x	y
toxin disposition	contagiousness

ido (http://purl.obolibrary.org/obo/ido/2017-11-03/ido.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
?x owl:disjointWith ?y .
OPTIONAL { ?x rdfs:label ?label }
} ORDER BY ?x
```

x	y
invasion disposition	disease

ido (http://purl.obolibrary.org/obo/ido/2017-11-03/ido.owl)

Active ontology x | Entities x | Individuals by class x | DL Query x | SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x owl:disjointWith ?y .
  OPTIONAL { ?x rdfs:label ?label }
  FILTER regex( ?label , "ster", "i" )
} ORDER BY ?x

```

x	y
---	---

ido (http://purl.obolibrary.org/obo/ido/2017-11-03/ido.owl)

Active ontology x | Entities x | Individuals by class x | DL Query x | SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT *
WHERE {
  ?x rdf:type ?y .
  ?y rdf:type ?Class .
  optional { ?x rdfs:label ?lab }
  optional { ?y rdfs:label ?lab }
  FILTER regex( ?lab , "universal", "i" )
}

```

x	y	Class
universal	denotator type	owl:Class

ido (http://purl.obolibrary.org/obo/ido/2017-11-03/ido.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT *
WHERE {
  ?x rdf:type ?y .
  ?y rdf:type ?Class .
  optional {?x rdfs:label ?lab}
  optional {?y rdfs:label ?labl}
  FILTER regex( ?lab , "placeholder", "i" )
}

```

x	y	Class
placeholder removed	obsolescence reason specification	owl:Class

ido (http://purl.obolibrary.org/obo/ido/2017-11-03/ido.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT *
WHERE {
  ?x rdf:type ?y .
  ?y rdf:type ?Class .
  optional {?x rdfs:label ?lab}
  optional {?y rdfs:label ?labl}
  FILTER regex( ?lab , "uncurate", "i" )
}

```

x	y	Class
uncurated	curation status specification	owl:Class

ido (http://purl.obolibrary.org/obo/ido/2017-11-03/ido.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?range ?domain
WHERE {
  ?x rdf:type owl:ObjectProperty
  optional { ?x rdfs:range ?range }
  optional { ?x rdfs:domain ?domain }
}

```

x	range	domain
has role	role	independent continuant
has participant	continuant	occurent
is concretized as	specifically dependent continuant	generically dependent continuant
bearer of	specifically dependent continuant	
realized in	process	realizable entity
function of		function
location of		
OBI_0001938	OBI_0001933	information content entity
OBI_0000295	OBI_0000011	
derives into		
has quality	quality	
located in	material entity	material entity
OBI_1110119	'independent continuant' and (not ('s)	'independent continuant' and (not ('s)
preceded by	occurent	occurent
RO_0002087		
IAO_0000039	IAO_0000003	
IAO_0000221	quality	IAO_0000109
OBI_0000643		

Execute

ido (http://purl.obolibrary.org/obo/ido/2017-11-03/ido.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?subject
WHERE { ?subject a owl:Class .
OPTIONAL { ?subject rdfs:label ?label } .
FILTER regex(?label , "infection inc", "i" )
} ORDER BY ?subject

```

subject

- infection incidence
- infection incidence proportion
- infection incidence rate

ido (<http://purl.obolibrary.org/obo/ido/2017-11-03/ido.owl>)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?subject
WHERE { ?subject a owl:Class .
OPTIONAL { ?subject rdfs:label ?label } .
FILTER regex( ?label , "drug su", "i" )
} ORDER BY ?subject

```

subject
drug susceptibility
drug susceptibility of infectious agent

ido (<http://purl.obolibrary.org/obo/ido/2017-11-03/ido.owl>)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?subject
WHERE { ?subject a owl:Class .
OPTIONAL { ?subject rdfs:label ?label } .
FILTER regex( ?label , "direct", "i" )
} ORDER BY ?subject

```

subject
direct host exposure to infectious agent
indirect

ido (http://purl.obolibrary.org/obo/ido/2017-11-03/ido.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x
WHERE {
?x rdf:type owl:ObjectProperty }

```

x
member of
regulates
realizes
is concretized as
positively regulates
function of
participates in
has participant
has member
disposition of
results in formation of
located in
2D boundary of
capable of
has output
role of
quality of
has quality

Execute

ido (http://purl.obolibrary.org/obo/ido/2017-11-03/ido.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x rdfs:subClassOf ?y .
  OPTIONAL { ?x rdfs:label ?label }
  FILTER regex( ?label , "process", "i" )
} ORDER BY ?x

```

x	y
process	occurrent
process boundary	occurrent
process profile	process

ido (http://purl.obolibrary.org/obo/ido/2017-11-03/ido.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x rdfs:subClassOf ?y .
  OPTIONAL { ?x rdfs:label ?label }
  FILTER regex( ?label , "virulence", "i" )
} ORDER BY ?x

```

x	y	label
obsolete_virulence	Obsolete Class	"obsolete_virulence"@en
obsolete_virulence factor	Obsolete Class	"obsolete_virulence factor"@en
virulence factor disposition	disposition	"virulence factor disposition"@en
virulence factor disposition	● inheres in' some 'molecular entity	"virulence factor disposition"@en
virulence	● inheres in' some 'infectious agent	"virulence"@en
virulence	quality	"virulence"@en
virulence factor	molecular entity	"virulence factor"@en

ido (http://purl.obolibrary.org/obo/ido/2017-11-03/ido.owl)

Active ontology x | Entities x | Individuals by class x | DL Query x | SPARQL Query x

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x rdfs:subClassOf ?y .
  OPTIONAL { ?x rdfs:label ?label }
  FILTER regex(?label, "anti", "i")
} ORDER BY ?x
```

x	y	label
_obsolete_antibiotic resistance	Obsolete Class	"_obsolete_antibiotic resistance"@en
_obsolete_antimicrobial	Obsolete Class	"_obsolete_antimicrobial"@en
_obsolete_antibiotic	Obsolete Class	"_obsolete_antibiotic"@en
_obsolete_antigenic variation	Obsolete Class	"_obsolete_antigenic variation"@en
antiseptic role	⦿ (inherits in some ('material entity' and ('has disposition' some 'antimicrobial 'antiseptic role'@en	"antiseptic role"@en
antiseptic role	role	"antiseptic role"@en

ido (http://purl.obolibrary.org/obo/ido/2017-11-03/ido.owl)

Active ontology x | Entities x | Individuals by class x | DL Query x | SPARQL Query x

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x rdfs:subClassOf ?y .
  OPTIONAL { ?x rdfs:label ?label }
  FILTER regex(?label, "antisept", "i")
} ORDER BY ?x
```

x	y	label
---	---	-------

OBI testi ar Protege

obi (http://purl.obolibrary.org/obo/obi/2022-07-11/obi.owl)

Active ontology x | Entities x | Individuals by class x | DL Query x | SPARQL Query x

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x rdfs:subClassOf ?y .
  OPTIONAL { ?x rdfs:label ?label }
  FILTER regex(?label, "Autoimmune", "i")
} ORDER BY ?x
```

x	y
---	---

obi (http://purl.obolibrary.org/obo/obi/2022-07-11/obi.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
?x rdfs:subClassOf ?y .
OPTIONAL { ?x rdfs:label ?label }
FILTER regex( ?label , "BLC", "i" )
} ORDER BY ?x

```

x	y

obi (http://purl.obolibrary.org/obo/obi/2022-07-11/obi.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
?x rdfs:subClassOf ?y .
OPTIONAL { ?x rdfs:label ?label }
FILTER regex( ?label , "Specimen", "i" )
} ORDER BY ?x

```

x	y

obi (http://purl.obolibrary.org/obo/obi/2022-07-11/obi.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x rdfs:subClassOf ?y .
  OPTIONAL { ?x rdfs:label ?label }
  FILTER regex( ?label , "Source", "i" )
} ORDER BY ?x

```

x y

obi (http://purl.obolibrary.org/obo/obi/2022-07-11/obi.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x rdfs:subClassOf ?y .
  OPTIONAL { ?x rdfs:label ?label }
  FILTER regex( ?label , "Patient", "i" )
} ORDER BY ?x

```

x y

obi (http://purl.obolibrary.org/obo/obi/2022-07-11/obi.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
?x owl:disjointWith ?y .
OPTIONAL { ?x rdfs:label ?label }
} ORDER BY ?x

```

x	y
---	---

obi (http://purl.obolibrary.org/obo/obi/2022-07-11/obi.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT *
WHERE {
?x rdf:type ?y .
?y rdf:type ?Class .
optional { ?x rdfs:label ?lab }
optional { ?y rdfs:label ?lab }
}

```

x	y	Class	lab
---	---	-------	-----

obi (http://purl.obolibrary.org/obo/obi/2022-07-11/obi.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?range ?domain
WHERE {
  ?x rdf:type owl:ObjectProperty
  optional { ?x rdfs:range ?range }
  optional { ?x rdfs:domain ?domain }
}

```

x	range	domain
part_of		
part_of		

obi (http://purl.obolibrary.org/obo/obi/2022-07-11/obi.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?subject ?label
WHERE { ?subject a owl:Class .
  OPTIONAL { ?subject rdfs:label ?label } .
  FILTER regex( ?label , "Ionize p", "i" )
} ORDER BY ?subject

```

subject	label

obi (http://purl.obolibrary.org/obo/obi/2022-07-11/obi.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?subject ?label
WHERE { ?subject a owl:Class .
  OPTIONAL { ?subject rdfs:label ?label } .
  FILTER regex( ?label , "Device", "i" )
} ORDER BY ?subject

```

subject	label

obi (http://purl.obolibrary.org/obo/obi/2022-07-11/obi.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?subject ?label
WHERE { ?subject a owl:Class .
OPTIONAL { ?subject rdfs:label ?label } .
FILTER regex( ?label , "Allel", "i" )
} ORDER BY ?subject

```

subject	label
---------	-------

obi (http://purl.obolibrary.org/obo/obi/2022-07-11/obi.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?label
WHERE {
?x rdf:type owl:ObjectProperty
OPTIONAL { ?x rdfs:label ?label } .
FILTER regex( ?label , "Has cat", "i" )
}

```

x	label
---	-------

obi (http://purl.obolibrary.org/obo/obi/2022-07-11/obi.owl)

Active ontology x | Entities x | Individuals by class x | DL Query x | SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?label
WHERE {
  ?x rdf:type owl:ObjectProperty
}

```

x	label
part_of	part_of

obi (http://purl.obolibrary.org/obo/obi/2022-07-11/obi.owl)

Active ontology x | Entities x | Individuals by class x | DL Query x | SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x rdfs:subClassOf ?y .
  OPTIONAL { ?x rdfs:label ?label }
  FILTER regex( ?label , "Secondary struc", "i" )
} ORDER BY ?x

```

x	y	label
---	---	-------

MFOMD testi ar Protege

< > MFOMD (http://purl.obolibrary.org/obo/MFOMD/2020-04-26/MFOMD.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?subject
WHERE { ?subject a owl:Class .
OPTIONAL { ?subject rdfs:label ?label } .
FILTER regex( ?label , "mental disease", "i" )
} ORDER BY ?subject
```

subject
mental disease
mental disease course
diagnosis of mental disease
manifestation of a mental disease

< > MFOMD (http://purl.obolibrary.org/obo/MFOMD/2020-04-26/MFOMD.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x
WHERE {
?x rdf:type owl:ObjectProperty }
```

x
realizes
manifestationOf
material basis of at all times
has occurrent part
part of occurrent
disposition of at all times
has disposition at some time
part of continuant at some time
has participant at all times
inheres in at all times
realized in
has participant at some time
has proper continuant part at all times
has continuant part at all times
has proper occurrent part
participates in at some time
isAbout

MFOMD (http://purl.obolibrary.org/obo/MFOMD/2020-04-26/MFOMD.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x rdfs:subClassOf ?y .
  OPTIONAL { ?x rdfs:label ?label }
}
FILTER regex( ?label , "mental disease", "i" )
} ORDER BY ?x

```

x	y
mental disease	disease
mental disease course	● has occurrent part some 'pathological mental process'
mental disease course	● realizes some 'mental disease'
mental disease course	disease course
diagnosis of mental disease	diagnosis
manifestation of a mental disease	● realizes some 'mental disease'
manifestation of a mental disease	● part of occurrent some 'mental disease course'
manifestation of a mental disease	manifestation of a disease

MFOMD (http://purl.obolibrary.org/obo/MFOMD/2020-04-26/MFOMD.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x rdfs:subClassOf ?y .
  OPTIONAL { ?x rdfs:label ?label }
}
FILTER regex( ?label , "narcolepsy", "i" )
} ORDER BY ?x

```

x	y
narcolepsy	sleep disorder

MFOMD (http://purl.obolibrary.org/obo/MFOMD/2020-04-26/MFOMD.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x rdfs:subClassOf ?y .
  OPTIONAL { ?x rdfs:label ?label }
}
FILTER regex( ?label , "Kleine-Levin", "i" )
} ORDER BY ?x

```

x	y
Kleine-Levin syndrome	recurrent hypersomnia

MFOMD (<http://purl.obolibrary.org/obo/MFOMD/2020-04-26/MFOMD.owl>)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x rdfs:subClassOf ?y .
  OPTIONAL { ?x rdfs:label ?label }

  FILTER regex( ?label , "mental process|", "i" )
} ORDER BY ?xx

```

x	y
pathological mental process	● manifestationOf some 'mental disorder'
pathological mental process	pathological bodily process
disposition to be agent of mental process	disposition
disposition to be agent of mental process	● disposition of at all times' some 'mental functioning relate
mental process	● 'has occurrent part' some consciousness
mental process	● realizes some 'mental disposition'
mental process	bodily process
mental process	bodily process
mental process	● realizes some 'mental disposition'
mental process	● 'has occurrent part' some consciousness

MFOMD (<http://purl.obolibrary.org/obo/MFOMD/2020-04-26/MFOMD.owl>)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x rdfs:subClassOf ?y .
  OPTIONAL { ?x rdfs:label ?label }

  FILTER regex( ?label , "seeing", "i" )
} ORDER BY ?xx

```

x	y
seeing	perception
seeing	● has occurrent part' some 'visual consciousness
seeing	● has occurrent part' some 'visual consciousness

MFOMD (http://purl.obolibrary.org/obo/MFOMD/2020-04-26/MFOMD.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x rdfs:subClassOf ?y .
  OPTIONAL { ?x rdfs:label ?label }
}
FILTER regex( ?label , "manifestation", "i" )

} ORDER BY ?xx

```

x	y
manifestation of a mental disease	● realizes some 'mental disease'
manifestation of a mental disease	● part of occurrent' some 'mental disease course
manifestation of a mental disease	manifestation of a disease
preclinical manifestation of a disease	manifestation of a disease
manifestation of a disease	quality
clinical manifestation of a disease	manifestation of a disease

MFOMD (http://purl.obolibrary.org/obo/MFOMD/2020-04-26/MFOMD.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
  ?x owl:disjointWith ?y .
  OPTIONAL { ?x rdfs:label ?label }
}

} ORDER BY ?xx

```

x	y
full-blown manic mood episode	hypomanic mood episode
full-blown manic mood episode	depressed mood episode
realizable entity	quality
specifically dependent continuant	generically dependent continuant
one-dimensional continuant fiat boundary	zero-dimensional continuant fiat boundary
one-dimensional continuant fiat boundary	two-dimensional continuant fiat boundary
material entity	immaterial entity
temporal region	spatiotemporal region
temporal region	process
disposition	role
continuant	occurrent
one-dimensional spatial region	three-dimensional spatial region
independent continuant	specifically dependent continuant
temporal region	process boundary
spatial region	continuant fiat boundary
independent continuant	generically dependent continuant
two-dimensional spatial region	three-dimensional spatial region
zero-dimensional spatial region	three-dimensional spatial region

Execute

MFOMD (<http://purl.obolibrary.org/obo/MFOMD/2020-04-26/MFOMD.owl>)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
?x owl:disjointWith ?y .
OPTIONAL { ?x rdfs:label ?label }

FILTER regex ( ?label , "disp", "i" )

} ORDER BY ?x

```

x	y
disposition	role

MFOMD (<http://purl.obolibrary.org/obo/MFOMD/2020-04-26/MFOMD.owl>)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?y ?label
WHERE {
?x owl:disjointWith ?y .
OPTIONAL { ?x rdfs:label ?label }

FILTER regex ( ?label , "contin", "i" )

} ORDER BY ?x

```

x	y
continuant	occurrent

MFOMD (http://purl.obolibrary.org/obo/MFOMD/2020-04-26/MFOMD.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT DISTINCT *
WHERE {

?ind a owl:NamedIndividual
optional { ?ind rdf:type ?type }
optional { ?type rdfs:label ?lab }
FILTER regex( ?lab , "denot", "i" )
}

```

ind	type
named class expression	denotator type
named class expression	denotator type
universal	denotator type
universal	denotator type
defined class	denotator type
defined class	denotator type

Execute

MFOMD (http://purl.obolibrary.org/obo/MFOMD/2020-04-26/MFOMD.owl)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT *
WHERE {
?x rdf:type ?y .
?y rdf:type ?Class .
optional { ?x rdfs:label ?lab }
optional { ?y rdfs:label ?labl }
FILTER regex ( ?lab , "uncura", "i" )
}

```

x	y	Class
uncurated	curation status specification	owl:Class

MFOMD (<http://purl.obolibrary.org/obo/MFOMD/2020-04-26/MFOMD.owl>)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT *
WHERE {
  ?x rdf:type ?y .
  ?y rdf:type ?Class .
  optional { ?x rdfs:label ?lab }
  optional { ?y rdfs:label ?labl }

  FILTER regex ( ?lab , "terms merged", "i" )
}

```

x	y	Class
terms merged	obsolescence reason specification	owl:Class

MFOMD (<http://purl.obolibrary.org/obo/MFOMD/2020-04-26/MFOMD.owl>)

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT ?x ?range ?domain
WHERE {
  ?x rdfs:domain ?domain
}

```

x	range	domain
BFO_0000062		occurent

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?x ?range ?domain
WHERE {
  ?x rdf:type owl:ObjectProperty
  optional { ?x rdfs:range ?range }
  optional { ?x rdfs:domain ?domain }
}

```

x	range	domain
realizes		
manifestationOf		
material basis of at all times		
has occurrent part		
part of occurrent		
disposition of at all times		
has disposition at some time		
part of continuant part at some time		
has participant at all times		
inheres in at all times		
realized in		
has participant at some time		
has proper continuant part at all times		
has continuant part at all times		
has proper occurrent part		
participates in at some time		
isAbout		
inheresOf		

Execute

Testa piemēri ar BTO<http://purl.obolibrary.org/obo/bto.owl>

Testa piemērs	Testa gadījums ar prasību
BTO_0005352 type Class	1-LN cell is a Class
BTO_0002603 type Class	11-9-1-4 cell is a Class
BTO_0001492 type Class	Limb is a Class
BFO_0000054 type Property	Realized in is a Property
RO_0003001 type Property	Produced by is a property
RO_0000053 type Property	Bearer of is a Property
BTO_0004468 subClassOf BTO_0000416	Lip epithelium is an epithelium
BTO_0001255 subClassOf BTO_0000452	Skin fibroblast is a fibroblast
BTO_0003622 subClassOf BTO_0000830	Mucosal mast cell is a mast cell
BTO_0000370 subClassOf BFO_0000050 some BTO_0000369	Egg white is a part of some egg
BTO_0003089 subClassOf BFO_0000050 some BTO_0002098	Promontorium tympani is a part of some tympanum
BTO_0005672 subClassOf RO_0002202 some BTO_0001321	LNcaP-C4-2 cell derives from or develops from some LNcaP cell
N/A	N/A
N/A	N/A
N/A	N/A
N/A	N/A
N/A	N/A
N/A	N/A
N/A	N/A
N/A	N/A
N/A	N/A

Testa piemēri ar CMOpurl.obolibrary.org/obo/cmo.obo

Testa piemērs	Testa gadījums ar prasību
CMO_0000714 type Class	Fractional change in blood vessel diameter per unit change in intravascular pressure is a Class
CMO_0000715 type Class	Calculated blood vessel distensibility is a Class
CMO_0001818 type Class	Calculated weight of islet beta cells in duodenal region of pancreas is a Class
N/A	N/A
N/A	N/A
part_of type Property	part_of is a Property
CMO_0000456 subClassOf CMO_0000227	Adipocyte measurement is a cell measurement
CMO_0001756 subClassOf CMO_0002163	Pituitary count is a pituitary measurement
CMO_0001707 subClassOf CMO_0001706	Benign liver tumor number is a liver tumor number
CMO_0000395 subClassOf part_of some CMO_0001333	Egg yolk weight, fowl is a part of some egg weight, fowl.
CMO_0000047 subClassOf part_of some CMO_0002334	Indirect plasma bilirubin level is a part of some total plasma bilirubin level
CMO_0000704 subClassOf part_of some CMO_0000700	Blood vessel distensibility is a part of some blood vessel resistance measurement
N/A	N/A
N/A	N/A
N/A	N/A
N/A	N/A
N/A	N/A
N/A	N/A
N/A	N/A
N/A	N/A
N/A	N/A

Testa piemēri ar OBI

<http://purl.obolibrary.org/obo/obi.owl>

Testa piemērs	Testa gadījums ar prasību
OBI_0000373 type Class	Ionize process is a Class
OBI_0000654 type Class	Device setting is a Class
OBI_0001352 type Class	Allele information is a Class
OBI_0000999 type Property	Has category label is a Property
OBI_0000846 type Property	Is member of organization is a Property
OBI_0000107 type Property	Provides service consumer with is a Property
OBI_0000896 subClassOf OBI_0000885	Secodary structure of RNA molecule is a secondary structure of sequence macromolecule
OBI_1110054 subClassOf OBI_0000031	Autoimmune disease is a disease
OBI_0002461 subClassOf IAO_0000098	BLC Format is a data format specification
OBI_0003070 subClassOf BFO_0000051 some OBI_0000684	Specimen collection protocol has part in some specimen collection objective
APOLLO_SV_00000522 subClassOf BFO_0000051 some IAO_0000051	(Source code repository is part of some source code module
OBI_0000093 subClassOf RO_0000052 some NCBITaxon_9606	Patient role inherits in some Homo sapiens
BFO_0000144 disjointWith BFO_0000182	Process profile is not history
OBI_0000678 disjointWith OBI_0000681	Portioning objective is not separation into different composition objective
BFO_0000038 disjointWith BFO_0000148	One dimensional temporal region is not zero dimensional temporal region
OBI_0000324 Type IAO_0000098	RDF is a data format specification
OBI_0000753 Type OBI_0000835	Waters is a manufacturer
OBI_0000979 Type OBI_0000963	ambidexterous is a categorical label
BFO_0000055 Domain BFO_0000015	Realizes domains process
RO_0000087 Range BFO_0000023	has role ranges role

Testa piemēri ar GENO

<http://purl.obolibrary.org/obo/geno.owl>

Testa piemērs	Testa gadījums ar prasību
GENO_0000921 type Class	biological sequence or set is a Class
GENO_0000702 type Class	biological sequence is a Class
GENO_0000022 type Class	obsolete genomic feature collection is a Class
IAO_0000219 type Property	denotes is a Property
IAO_0000424 type Property	expand expression to is a Property
IAO_0000425 type Property	expand assertion to is a Property
GENO_0000885 subClassOf GENO_0000823	diploptype is a allelic genotype
GENO_0000823 subClassOf GENO_0000897	allelic genotype is a genomic entity
GENO_0000611 subClassOf GENO_0000899	genomic background is a genomic genotype
GENO_0000823 subClassOf IAO_0000219 some GENO_000051	allelic genotype should denote some single locus complement
GENO_0000002 subClassOf GENO_0000683 some GENO_0000000	variant allele should vary with some reference allele
GENO_0000722 subClassOf GENO_0000783 some GENO_0000000	amino acid sequence should have sequence unit of some amino acid residue
BFO_0000002 disjointWith BFO_0000003	continuant cannot be occurrent
BFO_0000019 disjointWith BFO_0000017	quality cannot be realizable entity
GENO_0000815 disjointWith GENO_0000702	sequence feature location cannot be biological sequence
GENO_0000484 type IAO_0000102	exploratory term is data about an ontology part
IAO_0000410 Type IAO_0000409	universal is a denotator type
IAO_0000420 Type IAO_0000409	defined class is a denotator type
GENO_0000966 Domain GENO_0000815	has_interval domains sequence feature location
GENO_0000239 Range GENO_0000702	has_sequence ranges biological sequence

Testa piemēri ar HSO

<http://purl.obolibrary.org/obo/hso.owl>

Testa piemērs	Testa gadījums ar prasību
HSO_0000383 type Class	Extent of outbreak is a Class
BFO_0000140 type Class	Continuant fiat boundary is a Class
HSO_0000247 type Class	Fluid specimen is a Class
HSO_0000376 type Property	Available in language is a Property
HSO_0000307 type Property	Sampling stage is a Property
BSPO_0000108 type Property	Superficial is a Property
BFO_0000006 subClassOf BFO_0000141	Spatial region is a immaterial entity
BFO_0000008 subClassOf BFO_0000003	Temporal region is an occurrent
SIO_000666 subClassOf ENVO_01000658	Township is a dense settlement
UBERON_0000178 subClassOf BFO_0000051 some UBERON_0	Blood has part in some blood plasma
UBERON_0007779 subClassOf RO_0002494 some UBERON_0	Transudate is transformation of some blood plasma
UBERON_0000913 subClassOf BFO_0000050 some UBERON_0	Interstitial fluid is part of some tissue
BFO_0000016 disjointWith BFO_0000023	Disposition is not a role
OBI_0000070 disjointWith OBI_0000339	Assay is not planning
HSO_0000314 disjointWith HSO_0000318	Census sampling is not risk-based sampling
GAZ_00002938 Type NCIT_C25464	Belgium is a Country
GAZ_00002828 Type NCIT_C25464	Brazil is a Country
GAZ_00002845 Type NCIT_C25464	China is a Country
HSO_0000301 domain HSO_0000001	Target pathogen domains surveillance activity
BFO_0000055 range BFO_0000017	Realizes ranges realizable entity

Testa piemēri ar DOID

<http://purl.obolibrary.org/obo/dooid.owl>

Testa piemērs	Testa gadījums ar prasību
DOID_0070000 type Class	3-methylglutaconic aciduria type 8 is a Class
DOID_0050711 type Class	aceruloplasminemia is a Class
DOID_0050603 type Class	acheiropody is a Class
RO_0040035 type Property	disease relationship is a property
RO_0001000 type Property	derives from is a Property
RO_0004000 type Property	condition has genetic basis in is a property
DOID_0110009 SubClassOf DOID_13911	Achromatopsia 7 is a achromatopsia
DOID_13911 SubClassOf DOID_13399	achromatopsia is a color blindness
DOID_11776 SubClassOf DOID_1686	Absolute glaucoma is a glaucoma
DOID_0050593 SubClassOf RO_0004019 some SO_0000704	primary congenital glaucoma is a disease which has basis in some gene
DOID_0080041 SubClassOf IDO_0000664 some GENO_000014	hypochondropiasia has material basis in some autosomal dominant inheritance
DOID_0080038 SubClassOf RO_0002200 some HP_0003510	pyncnodysostosis has phenotype some severe short stature
DOID_0111134 DisjointWith DOID_0111138	congenital generalized lipodystrophy type 1 is not congenital generalized lipodystrophy type 4
DOID_0110957 DisjointWith DOID_0110961	Gauchers disease type 1 is not atypical Gauchers disease due to saposin c deficiency
DOID_0110937 DisjointWith DOID_0110940	autosomal dominant osteopetrosis 1 is not autosomal recessive osteopetrosis 8
N/A	N/A
N/A	N/A
N/A	N/A
N/A	N/A
N/A	N/A

Testa piemēri ar IDO

<http://purl.obolibrary.org/obo/ido.owl>

Testa piemērs	Testa gadījums ar prasību
IDO_0000484 type Class	infection incidence rate is a class
IDO_0000469 type Class	drug susceptibility is a class
IDO_0000594 type Class	direct host exposure to infectious agent is a class
RO_0001025 type Property	located in is a Property
RO_0002215 type Property	capable of is a Property
RO_0000081 type Property	role of is a Property
BFO_0000144 subClassOf BFO_0000015	Process profile is a process
IDO_0000466 subClassOf BFO_0000019	virulence is a quality
IDO_0000433 subClassOf BFO_0000023	antiseptic role is a role
IDO_0000482 subClassOf RO_0000052 some IDO_0000509	infection incidence proportion inheres in some organism population
IDO_0000635 subClassOf RO_0000052 some BFO_0000040	source of infection role inheres in some material entity
IDO_0000584 subClassOf RO_0000091 some IDO_0000426	cytotoxin has disposition some toxin disposition
IDO_0000426 disjointWith IDO_0000458	toxin disposition is not contagiousness
IDO_0000438 disjointWith OGMS_0000031	invasion disposition is not disease
IDO_0000449 disjointWith IDO_0000580	sterilizing immunity to infectious agent is not passive immunity to infectious agent
IAO_0000410 type IAO_0000409	universal is denotator type
IAO_0000226 type IAO_0000226	placeholder removed is obsolescence reason specification
IAO_0000124 type IAO_0000078	uncurated is curation status specification
BFO_0000062 domain BFO_0000003	preceded by domain occurrent
BFO_0000054 range BFO_0000015	realized in range process

Testa piemēri ar DIDEO

<http://purl.obolibrary.org/obo/dideo/release/2022-06-14/dideo.owl>

Testa piemērs	Testa gadījums ar prasību
DIDEO_00000149 type Class	cryopreserved human hepatocyte population is a class
DIDEO_00000162 type Class	human liver S9 fraction is a class
DIDEO_00000010 type Class	bioavailability is a class
DIDEO_00000027 type Property	organismally_begins_to_exist_during is a property
DIDEO_00000096 type Property	catalyzes a Phase I or Phase II enzymatic reaction involving is a property
DIDEO_00000090 type Property	inhibits-catalyzes metabolism of is a property
DIDEO_00000034 subClassOf BFO_0000016	absorbability is a disposition
DIDEO_00000173 subClassOf MOP_0000590	nifedipine dehydrogenation is a dehydrogenation
DIDEO_00000169 subClassOf CL_0000010	E. coli recombinant protein production host cell is a cultured cell
DIDEO_00000010 subClassOf RO_0000052 some DRON_0000005	bioavailability characteristic of some drug product
DIDEO_00000039 subClassOf RO_0000052 some OBI_0000427	primary total clearance enzyme role characteristic of some enzyme
DIDEO_00000020 subClassOf RO_0000057 some BFO_0000040	drug-drug interaction has participant in some material entity
BFO_0000016 disjointWith BFO_0000023	disposition is not role
BFO_0000017 disjointWith BFO_0000019	realizable entity is not quality
BFO_0000004 disjointWith BFO_0000031	independent continuant is not generically dependent continuant
OBI_0302900 type OBI_0302900	group randomization is group randomization
IAO_0000124 type IAO_0000078	uncurated is curation status specification
IAO_0000229 type IAO_0000225	term split is obsolescence reason specification
IAO_0000136 domain IAO_0000030	is about domain information content entity
RO_0000057 range BFO_0000002	has participant range continuant