

H.I. ESHONQULOV

**ONTOLOGIYALAR VA
BILIMLARNI IFODALASH,
AVTOMATLASHTIRILGAN
AXBOROT TIZIMLARINING
ONTOLOGIYASINI
QO'LLASH ASOSLARI**



ISBN 978-9943-7931-1-8



9 789943 793118



**O'ZBEKISTON RESPUBLIKASI OLIY VA O'RTA MAXSUS
TA'LIM VAZIRLIGI
BUXORO DAVLAT UNIVERSITETI**

ESHONQULOV HAKIM ILHOMOVICH

**ONTOLOGIYALAR VA BILIMLARNI
IFODALASH, AVTOMATLASHTIRILGAN
AXBOROT TIZIMLARINING
ONTOLOGIYASINI QO'LLASH ASOSLARI**

**“Durdona” nashriyoti
Buxoro – 2022**

UO'K 004:111

32.81

87.21

E 99

Eshonqulov, Hakim Ilhomovich

Ontologiyalar va bilimlarni ifodalash, avtomatlashtirilgan axborot tizimlarining ontologiyasini qo'llash asoslari [Matn] / H.I. Eshonqulov .-Buxoro: "Sadridin Salim Buxoriy" Durдона, 2022. -128 b.

KBK 32.81

87.21

Monografiyada ontologiyalar va bilimlarni ifodalash, avtomatlashtirilgan axborot tizimlarining ontologiyasini qo'llash asoslari, aqlli tizimlar uchun interfeys sifatida ontologiyalardan foydalanish, bilimlar omborining ontologiyasini yaratish, internet tarmog'ida ontologiyalardan foydalanib funksional tarmoqlarni ishlab chiqish asoslari taqdim etilgan.

Ushbu monografiya avtomatlashtirilgan tizimlarni ishlab chiquvchi dasturchi-injenerlar, axborot texnologiyalari yo'nalishida tahsil oluvchi talabalar va o'qituvchilar uchun mo'ljallangan.

Taqrizchilar:

Shafiyev T.R., BuxDU "Axborot texnologiyalari" fakulteti
"Axborot texnologiyalari" kafedrasi mudiri, t.f.f.d (Phd)

Mo'minov B.B., TATU "Informatika asoslari" kafedrasi mudiri
t.f.d prof.

Monografiya Buxoro davlat universitetining ilmiy-texnik kengashi tomonidan nashrga tavsiya etilgan (bayonnoma № 27, 28.09.2021)

ISBN 978-9943-7931-1-8

MUNDARIJA

Kirish	4
I bob. Ontologiyalar nima?	6
1.1. Ontologiyalar ta'rifi	6
1.2. Ontologiya turlari	17
II bob. Axborot tizimlari ontologiyalari	32
2.1. Ontologiyani qo'llashning turli jihatlari	32
2.2. OWL – WEB ontologiyalari uchun tavsif qilish tili	76
2.3. CYCL nima? CYCL uchun dasturlash tillari.....	108
Xulosa.....	120
Foydalanilgan adabiyotlar ro'yxati.....	123

KIRISH

O'tgan asrning so'nggi choragida axborot texnologiyalarining rivojlanishi ma'lumotlar bazasi texnologiyalarining rivojlanishi va ularning keng miqyosda joriy etilishi bilan tavsiflanadi. Bu asrning boshlarida xuddi shunday jarayonlar ontologik modellashtirish bilan bog'liq texnologiyalarni ishlab chiqishda kuzatildi. Ko'pgina tadqiqotchilar tashkilotda ontologiya kutubxonalaridan foydalanish to'g'risida allaqachon tushuncha hosil qilishgan.

Kelgusi o'n yilliklardagi axborot jarayonlari ma'lumotlar bazalaridan foydalanish hozirgi kabi odatiy holga aylanadi. Shu bilan birga, o'zbek tilida axborot muhandisligi ontologiyalari va ulardan foydalanish texnologiyalari haqida deyarli hech qanday adabiyot mavjud emas. Ushbu monografiya ushbu bo'shliqni to'ldirish uchun mo'ljallangan. Monografiya kompyuter fanida ontologiyalarni qo'llash bilan bog'liq tushunchalar, tillar va usullarning umumiy ko'rinishidir. Monografiyada ontologiyaga asoslangan axborot texnologiyalaridan foydalangan holda dunyodagi eng yirik tizimlar va loyihalar taqdim etilgan. Monografiya muallifining maqsadi informatika faniga qiziquvchilar va birinchi marta tanishish uchun ontologiyalarni axborot tizimlarida, Internetda, bilimlarni ifodalash va ulardan foydalanish vazifalarida foydalanish muammolariga aniq kirishni yozish edi. bu mavzu bilan. Afsuski, ontologik modellashtirish texnologiyalari haqida tayyor bo'lmagan o'quvchiga tushunarli tarzda gapirish juda qiyin.

Gap shundaki, bilimlarni tavsiflash uchun ko'plab usullar qo'llaniladi, bu esa ancha keng qamrovli dastlabki o'rganishni talab qiladi. Masalan, ontologik qurilish tizimlarining eng mashhur joriy ilovalari birinchi va ikkinchi tartibli predikatlar hisob tillariga asoslangan. Shubhasiz, ontologiyani qurishning u yoki bu tizimi qanday ishlashini tushunish uchun, avvalo, ushbu tizimning markazida nima yotganini tushunish kerak. Tez tushunish uchun yana bir to'siq ko'proq asosiy hisoblanadi. Bu bilim ontologiyasini qurish bu bilimlarni modellashtirish jarayonidan boshqa narsa emasligidan iborat, ya'ni hal qilinayotgan muammo dunyosi modelini qurish. Modelni yaratish murakkab va mahoratli jarayondir. Yaxshi model, bir tomondan, muammo dunyosini to'g'ri aks ettirish uchun Yetarlicha

umumiy bo'lishi kerak, ikkinchi tomondan, muammoni hal qilish uchun zarur bo'lgan barcha tafsilotlarni o'z ichiga olishi kerak. Tayyor bo'lmagan o'quvchida model nima uchun bunday tarzda qurilgan, nega ba'zi tushunchalar modelda tavsiflangan, boshqalari muhim bo'lib tuyulgan, aksincha, ko'rib chiqilmagan va hokazo kabi savollarga muqarrar. Muayyan qarorning sabablarini to'g'ri tushunish uchun mavzuga biroz sho'ng'ish kerak.

Modelni yaratish jarayoni kontseptualizatsiya deb ataladi. Ushbu atamaning o'zi modelni yaratish jarayonida tushunchalar - model asosida yotqizilgan tushunchalar va bu tushunchalar o'rtasidagi munosabatni ajratib ko'rsatishini ko'rsatadi. O'quvchi, shuningdek, kontseptsiya nima ekanligini va uning rasmiy ravishda qanday ifodalanishi haqida bir oz tushunchaga ega bo'lishi kerak. Bu savollarga oydinlik kiritish uchun falsafaga murojaat qilish kerak. Shunday qilib, Monografiya, shuningdek, kontseptualizatsiya jarayonining asosi bo'lgan falsafiy g'oyalar haqida ma'lumot berishi kerak. Va nihoyat, shuni unutmash kerakki, Monografiya ontologik qurilish tizimlarining eng mashhur ilovalari haqida fikr berishi kerak, ya'ni qaysidir ma'noda ushbu tizimlarning umumiy ko'rinishini taqdim eting. Har bir bunday tizimni bir nechta paragraflarda tavsiflash mumkin, ammo afsuski, bunday kichik hajmdagi ma'lumotlar o'quvchiga bunday tizim mavjudligini tushunishdan boshqa hech narsa bermaydi. O'quvchiga tizimni yaxshiroq tushunish uchun butun bob kerak. Shunday qilib, tizimga botirish kerak. Monografiya ontologiya kontseptsiyasi bilan bog'liq g'oyalarga kirishdir va shu bilan birga ontologiyalarni qurish tizimlarining eng mashhur amalga oshirilishiga umumiy nuqtai nazardir. Monografiya matnini hech qanday tayyorgarliksiz o'qish mumkin.

Birinchi bob semantik texnologiyalar va axborot tizimlarining oddiy taqdimotidir. Shuningdek, birinchi bobda ontologiyalarning har xil turlariga misollar keltirilgan. Monografiyani o'qishni ushbu bobdan boshlash kerak. Qolgan bo'limlarni istalgan tartibda o'qish mumkin. Monografiyadan ontologik qurilish tizimlari bo'yicha ma'lumotnoma sifatida ham foydalanish mumkin.

I BOB. ONTOLOGIYALAR NIMA?

1.1. Ontologiyalar ta'rifi

Ushbu monografiya muhandislikda "ontologiya" tushunchasini qo'llashning turli usullarini tizimli ravishda tavsiflashga urinishdir. Ontologiya endi moda atamasi bo'lib, u inson faoliyatining ko'plab sohalarida (noto'g'ri yoki yo'q) qo'llaniladi. "Ontologiya" so'zini nafaqat ilmiy ishlarda, balki gazeta maqolalarida yoki teleko'rsatuvlarda ham uchratish mumkin. Umuman olganda, bir so'z bilan aytganda: "eshitildi". Ammo, har qanday yangi tushuncha kabi, "ontologiya" atamasi hali aniq talqinni olmagan. Bu ish ushbu bo'shliqni to'ldirishga urinishdir, garchi unda ko'rsatilgan yondashuvlarni yakuniy haqiqat deb hisoblash kerak, deb da'vo qilmasa ham. Monografiyada muallif faol a'zosi bo'lgan ilmiy guruhdagi tadqiqot jarayonida shakllangan qarashlar tavsiflangan. Muhandislikda jarayon modellarini qurish uchun odatda "ontologik" usullar qo'llaniladi. Jarayonning muhandislik modeli ontologiyadir. Aniqroq aytganda, ontologiya bu modelning tavsifidir.

Bunday model tavsiflari odatda rasmiydir, ya'ni konstruksiyalari har doim to'g'ri va bir ma'noda talqin qilinadigan, shu maqsadda maxsus ishlab chiqilgan tilda qilingan. Bu sizga tavsifni keyinchalik boshqa vazifada ishlatish uchun kompyuter xotirasida saqlash yoki tavsifda mantiqiy qarama-qarshiliklar mavjudligini kompyuter bilan tekshirish imkonini beradi. Aslida, modellarning rasmiy tavsiflarini avtomatlashtirilgan almashish vazifasi ontologiyani qo'llash sohasidagi tadqiqotlarni boshlash uchun asosiy turtki bo'ldi. Bunday qiyinchiliklar yuzaga keladigan muhandislik sohalarini ko'rib chiqing. Birinchi marta modelning rasmiy tavsifini qurish muammosi bilimlar bazalarini boshqarish uchun dasturiy ta'minot tizimlarida paydo bo'ldi. Bilimlar bazasi bu maqsad uchun maxsus ishlab chiqilgan dastur tomonidan boshqariladigan faktlarning tizimli to'plamidir. Baza ekspert tizimi sifatida amalga oshiriladi, ya'ni muayyan mavzu bo'yicha mutaxassis-mutaxassis vazifasini bajaradigan dastur. Misol uchun, tibbiyotda ekspert tizimlari hozir juda mashhur. Bunday bilimlar bazasida bir kishi eslay olmaydigan juda ko'p tibbiy faktlar mavjud. Ekspert tizimi qon tahlillari natijalarini kirish sifatida qabul qilishi mumkin va dastur o'z bilimlari bazasi faktlarini taqqoslab,

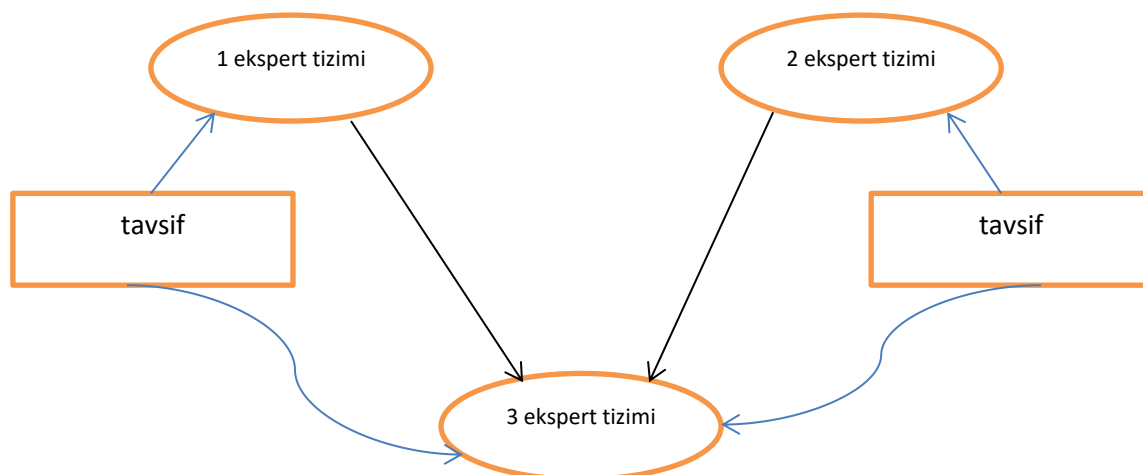
ushbu tahlillar natijalariga ko'ra bemorga tashxis qo'yishi mumkin. Ekspert tizimini unga yangi faktlar qo'shish orqali qayta tayyorlash mumkin. Aytaylik, turli ishlab chiqaruvchilar tomonidan ishlab chiqarilgan va turli tibbiyot muassasalarida joylashgan bunday ikkita ekspert tizimi mavjud va Sog'liqni saqlash vazirligi mavjud ikkitasi asosida yangi ekspert tizimini yaratishga qaror qildi. Agar mutaxassislar mashinalar emas, odamlar bo'lganida, hammasi oddiy bo'lardi: ular uchrashib, bir-birlariga bilganlarini aytib berishardi. Mashinalar bilan vaziyat boshqacha - ular bir-biriga hech narsa aytolmaydi. Shu sababli, har bir ekspert tizimining bilimlar bazasining mazmunini rasmiy ravishda tavsiflash kerak, shunda bu tavsif ularning kombinatsiyasini amalga oshiradigan dastur uchun tushunarli bo'ladi. Bu yerda quyidagi qiyinchiliklar yuzaga keladi:

1. Uchta ishtirokchi dasturning har biri tushunadigan tilni ishlab chiqish kerak. Bu til bilimlar bazalarining (ontologiyalarning) tavsiflari mashinalar tomonidan bir ma'noda talqin qilinishi uchun Yetarlicha rasmiy bo'lishi kerak.

2. Ushbu tilda bilimlar bazasining mazmuni tavsifini tuzing. Ko'rinib turibdiki, inson bunday tavsifni bera olmaydi, chunki u ekspert tizimining butun bilimlar bazasining mazmunini yodda tuta olmaydi. Bu shuni anglatadiki, tavsif dastur tomonidan amalga oshirilishi kerak.

3. Rivojlangan tilda tuzilgan bilimlar bazasi tavsifiga ko'ra, uchinchi ekspert tizimining bilimlar bazasiga yangi faktlar qo'shing. Dastur ham buni amalga oshirishi kerak.

Muhandislik fanining an'analariga to'liq mos ravishda, bilimlarni tavsiflash uchun tilni standartlashtirish g'oyasi paydo bo'ldi, shunda uni rivojlantirishga boshqa kuch sarflanmaydi. Keyin yuqoridagi ro'yxatning ikkinchi va uchinchi bandlari shunchaki bilimlar bazasining ichki tilidan ushbu tilga tarjimonlar yozishga, tavsiflash tilidan esa ekspert tizimida bilimlarni ifodalash tiliga qisqartiriladi. Bu amalga oshirildi (bilim almashish tili KIF — Knowledge Interchange Format deb ataladi), lekin bu qanday amalga oshirilganligi haqida keyinroq ikkinchi bobda gaplashamiz.

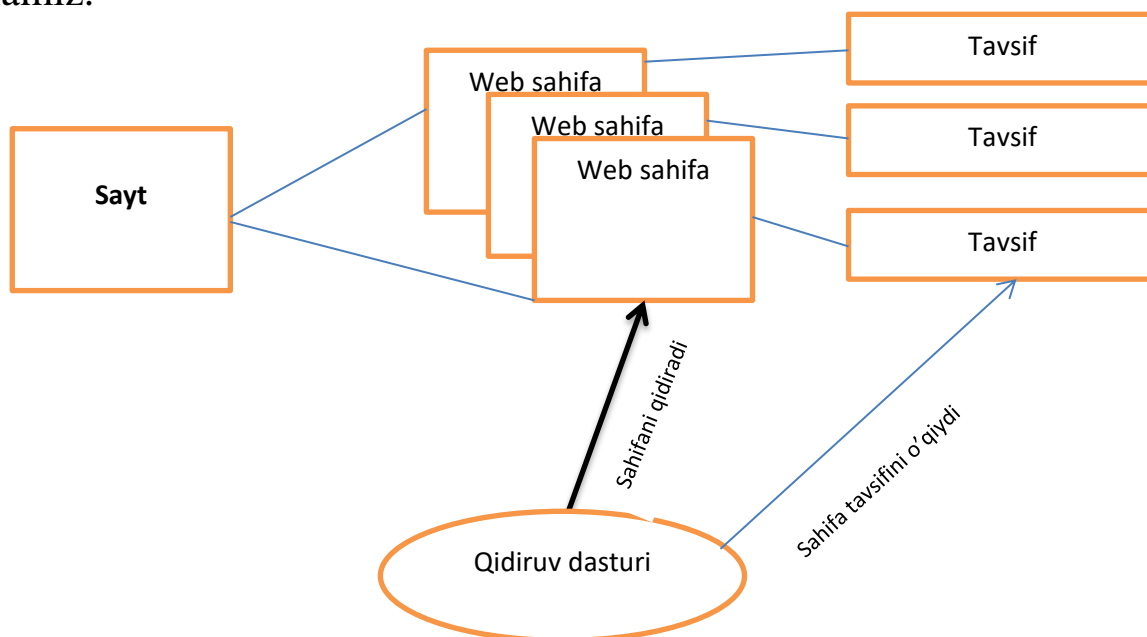


Rasm 1.1: Ekspert tizimlari orasidagi bilimlar almashinuvi

Model tavsiflarini qurish zarurati foydalanuvchi so'rovlari bo'yicha Web-sahifalarni topish muammosida ham paydo bo'ldi. Internetning dastlabki kunlarida qidiruv tizimlari Internetda so'rov matnini o'z ichiga olgan sahifalarni izlash bilan shug'ullangan. Misol uchun, agar foydalanuvchi "Eshmat" so'rovini kiritgan bo'lsa, u holda qidiruv dasturi sahifa matnlarida "Eshmat" so'zini qidirib topdi va ushbu so'zni o'z ichiga olgan sahifalarni yetarli miqdorda ko'rsatdi. Biroq, "Eshmat" so'zi noaniq talqinga ega.

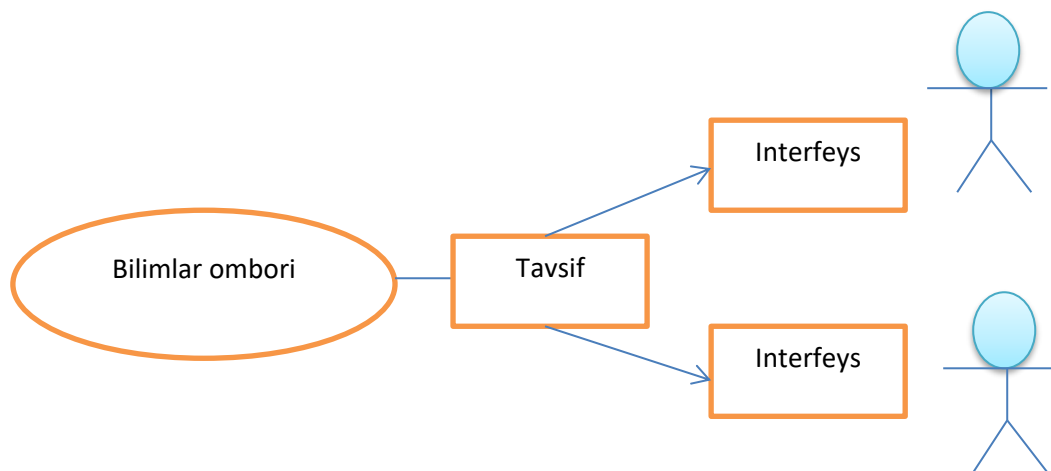
Misol uchun, bu mashhur shaxsning ismi bo'lishi mumkin: "Eshmat" yoki "Eshmat Toshmat", bu holda, bu odamlar haqidagi sahifalarni ko'rsatish kerak. Bundan tashqari, bu Eshmat shahrining nomi bo'lishi mumkin, keyin bu shahar haqidagi sahifalarni ko'rsatish kerak. Yetarlicha aqlli qidiruv dasturi "Eshmat" so'zining turli ma'nolarini "bilish" uchun yetarlicha katta faktlar bazasiga ega bo'lishi mumkin. Biroq, saytlar yaratuvchilari qidiruv dasturlariga o'z sahifalari mazmuni tavsifini bermaguncha, bu bunday dasturga hech narsa bermagan bo'lardi. Aytaylik, Eshmat shahri haqidagi sayt shunday tavsifga ega. Keyin, qidiruv dasturi foydalanuvchidan so'rovda "Eshmat shahri" degan ma'noni anglatadimi yoki yo'qligini so'rashi mumkin va agar u rozi bo'lsa, topilgan sahifalar ro'yxatida birinchi navbatda ushbu saytning sahifasini tavsifi bilan birga ko'rsatishi mumkin. Web-sahifalar mazmunining tavsiflari bu sahifalarning ontologiyasi hisoblanadi. Sahifalar o'z mazmuni tavsifiga ega bo'lgan Internet semantik veb deb ataladi. Ontologiyalarni yaratish uchun dasturlarni qidirish uchun tushunarli bo'lgan maxsus til kerak. Bunday til yaratilgan va OWL (Web Ontology Language) deb ataladi. Ushbu til va Web-sahifalarning

tavsiflarini qurish muammolarini ikkinchi bobda batafsil muhokama qilamiz.



Rasm 1.2: Qidiruv dasturlari uchun web sahifa tavsifi

Bilimlar tavsifini qurish har qanday hajmli bilimlarni boshqarish tizimida amalga oshirilishi kerak. Agar bilimlar bazasini ishlab chiquvchilar uni yangi faktlar bilan to'ldirishni rejalashtirsalar, buning uchun birinchi navbatda ushbu bilimlar bazasining mazmunini tizimlashtirish muhimdir. Bilimlar bazasida bo'limlarni tanlash, ularning mazmunini, shuningdek, turli bo'limlarni bir-biri bilan bog'lash usullarini tavsiflash kerak. Bunday holda, foydalanuvchilar kerakli joyda tizimga yangi bilimlarni qo'shadilar. Misol uchun, agar tizim tibbiyot va biokimyoviy jarayonlar bo'yicha bilimlarni o'z ichiga olsa, yangi dori haqida ma'lumotni kiritish paytida, bu preparatni qanday alomatlar uchun ishlatish kerakligi haqidagi bilimni qanday kimyoviy moddalar haqidagi bilim bilan bog'lash kerak bo'ladi. uning tarkibida dori bor va bu dorini qabul qilish qanday fiziologik ta'sir ko'rsatadi. Buning uchun bilimlar bazasining ontologiyasini qurish (ya'ni, uning tavsifi) va bu ontologiyada turli bo'limlarni bir-biri bilan bog'lash kerak. Keyin foydalanuvchi ma'lumotni kiritishda kerakli maydonlarni to'ldiradi va qo'shilgan bilimlar izchil va mantiqiy sxemaga bog'lanadi. Bu yerda bilimlarni tavsiflash uni tizimlashtirishda muhim rol o'ynashini ko'ramiz. CYC bilimlar bazasidan misol sifatida foydalanib, biz katta hajmli bilim bazalari qanday tashkil etilganligini ko'rib chiqamiz.



Rasm 1.3: *Bilimlar omborining interfeysi sifatida ontologiyalardan foydalanish.*

Bilimlarni tizimlashtirish umuman oson ish emas. To'g'ridan-to'g'ri ma'lumotlar bazasida mavjud bo'lgan bilimlardan tashqari, umumiy bilimlar ham kerak. Inson bunday bilimlar haqida hatto o'ylamaydi, ular odatiy hol sifatida qabul qilinadi. Masalan, odam odamlarning teskari va oyoqlarini erga qo'yib yurishini va umuman, ularning yurishini tortishish kuchining mavjudligi oqibati deb biladi. Odamning xayoliga devordan o'tish mumkinligi hech qachon kelmaydi, chunki u devor o'tib bo'lmaydigan narsa ekanligini biladi. Faqat ma'lum bilimlarni o'z ichiga olgan ma'lumotlar bazasida inson uchun tushunarli bo'lgan barcha tushunchalar mavjud emas. Shuning uchun dunyoning asosiy xossalarini - umumiy bilim deb ataladigan narsalarni rasmiy ravishda tavsiflash va bu bilimlarni turli xil maxsus bilimlar asoslarida qo'llash g'oyasi tug'ildi. Ushbu tavsiflar odatda yuqori darajadagi ontologiyalar deb ataladi. Oddiy bilimlar bazasi ko'pincha uchta bilim darajasini o'z ichiga oladi:

1. Yuqori darajadagi ontologiyada tavsiflangan umumiy yoki mavhum bilimlar.
2. Muayyan fan sohasiga oid bilimlar (elementga xos bilim). Masalan, geografiya, siyosatshunoslik yoki tibbiyot bilimlari.
3. Foydalanuvchilar yoki dasturiy agentlar tomonidan bilimlar bazasiga qo'shilgan maxsus bilimlar.

Bilimlarning darajalarga bo'linishi odatda insonga xosdir, biz bu haqda quyida ikkinchi bobning boshida batafsil gaplashamiz (2.1-bo'limga qarang). Umumiy bilimlarni tavsiflashning turli xil yondashuvlari mavjud, biz ushbu yondashuvlarni ikkinchi bobning

2.2-bo'limida batafsil ko'rib chiqamiz. Endi ontologiyalar muhandislikning ko'plab sohalarida qo'llaniladi. Yuqorida biz bir nechta misollarni ko'rib chiqdik, unda bilimlarni tavsiflamasdan hech narsa qilish mumkin emas. Biroq, mashinalar uchun bilimlarni rasmiy tavsiflashning ahamiyati hali to'liq tushunilmagan va shuning uchun bu sohalarida ontologik yondashuvlardan foydalanish kutilmagan ko'rinadigan sohalar ham mavjud. Masalan, tilshunoslik sohasini ko'rib chiqaylik. Bizning asrimizda tilni bilish ko'plab darsliklarda rasmiy ravishda tavsiflangan va turli mutaxassislar tomonidan muvaffaqiyatli qo'llaniladi. Ammo ma'lum bo'lishicha, bu sohada ham bilimlarni mashinalar uchun qulay shaklda tavsiflash zarur bo'lib qolgan. Shunday qilib, bir tildan ikkinchi tilga kompyuter tarjimonini yaratish katta ahamiyatga ega vazifa ekanligi ma'lum.

Ushbu muammoni hal qilish insoniyatning ko'p kuchini tejaydi. Har yili dunyoda ko'plab tarjimonlar tayyorlanmoqda, davlatlar bunday mutaxassislarni tayyorlash uchun juda katta mablag' sarflaydilar. Agar mashina tarjimoni yaratilgan bo'lsa, bu mablag'lar boshqa sohalarida ham mutaxassislar tayyorlashga yo'naltirilishi mumkin edi. Avtomatik tarjima muammosini hal qilish uchun tillar haqidagi bilimlarni rasmiy tavsiflash zarurati ayon bo'ldi. Mashina tarjimasini dasturlarining oldingi avlodlarini ishlab chiquvchilar bu dasturlarni oddiy ilovalar sifatida yaratishga harakat qilishgan. Dasturlash modeli yaratildi, tarjima mantiq'i dastur kodi bilan tavsiflandi va lug'at odatda ma'lumotlar bazasida saqlanadi. Ammo ma'lum bo'lishicha, til haqidagi bilim shunchalik kattaki, uni to'g'ridan-to'g'ri dastur modelida kodlash juda qiyin. Bundan tashqari, til haqidagi bilimlarni rasmiylashtirishda qiyinchiliklar mavjud - bu bilimlarning ko'plab nozik tomonlari hali rasman tavsiflanmagan. Ko'rsatilgan qiyinchiliklar bilan bog'liq holda, ushbu tavsifni turli dasturlarda, shu jumladan mashina tarjimasini dasturlarida ishlatish uchun mashina formatida til haqidagi bilimlarning tavsifini yaratish g'oyasi paydo bo'ldi. Bunday loyihalar mavjud, masalan, WordNet deb nomlangan ingliz tilidagi bilimlarni tavsiflash loyihasi mavjud. Umuman olganda, muallifning fikricha, muhandislikning turli sohalarida bilimlarning rasmiy tavsifini joriy etish jarayoni o'tgan asrning 70-80-yillarida ma'lumotlar bazalari bilan sodir bo'lgan jarayonga o'xshaydi. O'tgan asrning 60-yillarida ma'lumotlarning tavsifini ma'lumotlarning o'zidan ajratish zarurati amalga oshirildi.

Ma'lumotlar tavsifi ma'lumotlar bazasi sxemasi (kontseptual sxema) deb ataladi. Bunday tavsiflarning rasmiy nazariyasi yaratildi va keyin bunday tavsiflar bilan ishlay oladigan ilovalar yaratildi - ma'lumotlar bazasi serverlari. Shunga o'xshash jarayonlar, faqat bilimlarni tavsiflash bilan bog'liq holda, bizning davrimizda kuzatiladi. Ma'lumotlar bazalarining joriy etilishi ma'lumotlarni qayta ishlash bilan bog'liq turli sohalarda odamlarni mashinalar bilan almashtirishga va shu bilan ishlov beriladigan ma'lumotlar hajmini sezilarli darajada oshirishga imkon yaratgani kabi, ontologiyalarni joriy etish mashinalarga insoniy mehnat bilan bog'liq bo'lgan ba'zi intellektual faoliyat sohaslarida ishlash imkonini beradi.

"Ontologiya" atamasi yangilik emas va falsafada 600 yilga yaqin qo'llanilgan. Bu atama Marburg shahridan bo'lgan o'rta asr faylasufi Rudolf Goklenius tomonidan kiritilgan. 1613-yilda Goklenius mashhur «Falsafiy lug'at»ni nashr etdi va u yerda «antinomiya» atamasi bilan birga «ontologiya» atamasi ham taklif qilindi. 1656 yilda bu atama ("ontosofy" ekvivalent atamasi shaklida) nemis faylasufi Iogann Klauberg tomonidan o'z ishida "metafizika" tushunchasiga ekvivalent sifatida ishlatilgan. Keyinchalik nemis faylasuflari Kristian fon Volf "ontologiya" va "metafizika" atamalarining semantikasini bir-biridan ajratib oldilar. Endilikda "metafizika" atamasi falsafaning voqelikning boshlang'ich tabiatini o'rganuvchi bo'limini belgilash uchun yoki umuman olganda, fizikdan tashqariga chiqadigan barcha voqelikni belgilash uchun ishlatiladi.² Ontologiya metafizikaning torroq sohasini anglatib, unga bag'ishlangan. ob'ektlarning (tushunchalarning) bir-biri bilan munosabatidagi ma'nolariga ... "Ontologiya" so'zi qadimgi yunon tilining ikki ildizidan olingan: *οντος* — mavjud bo'lgan narsa (mavjud) va *λογος* — ta'limot (fan). Zamonaviy falsafada bu atama ikki xil tushuniladi. Bir tomondan, bu atama borliq haqidagi ilmni anglatadi. Boshqa tomondan, "ontologiya" atamasi ikkinchisining poydevorida yotgan deb hisoblangan borliqning ba'zi tomonlarini bildiradi. Bu yerda asosiy ta'rif:

Ta'rif 1.1: Ontologiya (falsafada) falsafaning borliq muammolarini o'rganuvchi bo'limidir.

O'quvchi, ehtimol, muhandislikda "ontologiya" atamasini tushunish falsafiy tushunchadan farq qilishiga allaqachon ishonch hosil qilgan. Texnikada "ontologiya" tushunchasi quyidagicha

shakllangan. Birinchidan, dasturiy ta'minot tizimining modeli tomonidan berilgan bilimlarni ushbu dasturiy ta'minot tizimining xatti-harakatlaridan ajratish zarurati amalga oshirildi.

Keyinchalik, ushbu tushuncha asosida dasturiy ta'minot tizimlari bilimlarining rasmiy tavsiflarini qurish zarurligi to'g'risida fikr shakllantirildi. Keling, bu qanday sodir bo'lganini batafsil ko'rib chiqaylik. Bilim tavsifini bilimning o'zidan ajratish zarurligini anglash o'tgan asrning 80-yillari oxirida paydo bo'la boshladi. Ushbu ehtiyojni amalga oshirish bilan birga, kompyuterning intellektual tizimlariga nisbatan "bilim" tushunchasini qayta ko'rib chiqish boshlandi.

Bungacha bilim funktsional nuqtai nazardan ko'rib chiqilgan:

Bilim - bu (dasturiy ta'minot) agentga tushuntirilishi mumkin bo'lgan narsa, shuning uchun uning xatti-harakati oqilona xatti-harakatlar tamoyillariga muvofiq hisoblanishi mumkin.

Bu fikr mantiqiy ko'rinadi. Agar inson uchun bilim birinchi navbatda haqiqatga yoki agar xohlasangiz, bu odamning dunyo (yoki uning bir qismi) haqidagi g'oyalarining to'liqligiga ishonish bo'lsa, unda mashina uchun e'tiqod tushunchasi mutlaqo qo'llanilmaydi. Shu sababli, bilim dasturiy ta'minot tizimini oqilona va bashorat qilinadigan harakatga keltiradigan ma'lumot sifatida qaraladi. Ma'lum bo'lishicha, bu ma'lumot qandaydir mutaxassis - bilimga ega bo'lgan odamdan keladi. Keyinchalik, bilimni ma'lum bir mutaxassisning ruhiy holati modeli emas, balki dunyo haqidagi g'oyalar modeli sifatida ko'rib chiqish taklif qilindi. Ushbu yondashuvga ko'ra, kompyuter tizimi haqidagi bilimlar ob'ektiv voqelikning modeli, aniqrog'i, uning predmet sohasi deb ataladigan bir qismidir. Ammo, keyin dunyoni tavsiflash modelini dasturiy ta'minot agentining xatti-harakatidan ajratish mumkin.

"Ontologiya" atamasi birinchi marta Tomas Gruber asarida paydo bo'lgan. Gruber bilimlar bazalarining dasturiy tizimlarining bir-biri bilan o'zaro ta'siri mexanizmini yaratish bo'yicha muhandislik muammosini hal qildi. Ushbu muammoni hal qilish uchun quyidagi usullar taklif qilingan:

- Bilimlarni boshqarish tizimida deklarativ deb ataladigan bilim darajasini ajratib ko'rsating. Bu ma'lum bir tizimda amalga oshiriladigan dunyo nimadan iboratligi haqidagi bilimdir, ya'ni deklarativ bilim - berilgan dunyoning tavsifi. Deklarativ bilimlardan

farqli o'laroq, protsessual bilimlarni kiritish mumkin, masalan, dunyoni tavsiflash asosida uning ayrim xususiyatlari bo'yicha mantiqiy xulosalar chiqarish. Boshqacha aytganda, bilimni qayta ishlashdan ajratish taklif qilindi.

- Bilimlar bazasi boshqa bilimlarni boshqarish dasturlariga o'zining deklarativ bilimlarining tavsifini beradi. Bundan tashqari, bunday tavsif odamga ham, mashinaga ham tushunarli bo'lishi kerak va shuning uchun u ikki shaklda chiqariladi:

1. *Predikat mantiqining odatiy tilida bilimlarning tavsifi bo'lgan kanonik shakl.*

2. *Ontologiya shaklida, xuddi o'sha paytda bo'lgani kabi, Gruber, ya'ni atamalarning tavsiflari (sinflar, munosabatlar, doimiylar va boshqalar) va bu atamalarni bir-biri bilan bog'laydigan ta'riflar shaklida.*

- Tanlangan tavsiflar asosida turli bilim bazalarida foydalanish mumkin bo'lgan ontologiya kutubxonalarini yarating.

Shunday qilib, T.Gruber texnikada "ontologiya" atamasining muallifi hisoblanadi. Bilim tavsifini qurish vazifasi juda aniq. Gruber hatto ushbu vazifa uchun alohida "kontseptsiya spetsifikatsiyasi" atamasini ajratib ko'rsatdi. Bu yerda "kontseptuallashtirish" "odamlar tomonidan qandaydir maqsadga erishish uchun foydalaniladigan dunyoning mavhum, soddalashtirilgan ko'rinishi" degan ma'noni anglatadi. Kontseptualizatsiya muammosining o'ziga xos xususiyati shundaki, dasturiy ta'minot tizimlari o'rtasida bilim almashish uchun ularning kontseptualizatsiyasini aniq ko'rsatish kerak, ya'ni bu bilimlarning tavsifini qurish, bundan tashqari, boshqa tizimlar tomonidan tushuniladigan darajada rasmiy. Ushbu spetsifikatsiyaning natijasi Gruber tomonidan "ontologiya" deb nomlangan.

Shunday qilib, muhandislik ontologiyasi kontseptsiyasini ma'lum bir kontseptualizatsiyaning spetsifikatsiyasi (rasmiy tavsifi) sifatida aniqlash mumkin (o'rganilayotgan muammoning mavzu sohasini ushbu muammo uchun zarur bo'lgan tarzda taqdim etish). Agar bu yerda spetsifikatsiya bilan hamma narsa ko'proq yoki kamroq aniq bo'lsa, kontseptualizatsiya nima to'liq aniq emas. Gruber kontseptualizatsiya sinflar va atributlar nuqtai nazaridan amalga oshiriladi, deb hisoblagan. O'rganilayotgan muammoning dunyosi sinflar, ularning xususiyatlari (atributlari) va o'ziga xos ob'ektlar -

sinflar misollari bilan tavsiflangan tushunchalar shaklida ifodalanadi. Masalan, avtomashinalar haqidagi ma'lumotlarni o'z ichiga olgan dasturiy ta'minot tizimini loyihalashda avtomobil markalarini (Toyota, Lada, Nissan va boshqalar) ifodalovchi sinflarni kiritish va tegishli sinflarning namunalari sifatida aniq avtomobillarni ko'rib chiqish tabiiydir. Ushbu yondashuv mashhur bo'lib, ob'ektga yo'naltirilgan dasturlash tamoyillariga asoslanadi.

Muallif kontseptualizatsiya ta'rifini faqat sinflar va ularning atributlari ko'rinishidagi vakillikka qisqartirish biroz tor deb hisoblaydi. 2.1-bo'limda kontseptsiyaning tipik vakillari deb ataladigan prototiplar yordamida tushunchalarni aniqlash misoli keltirilgan. Shunday qilib, "asbob" tushunchasi uchun "bolg'a" prototipi, "meva" tushunchasi uchun esa "olma" prototipi mavjud. Albatta, ba'zi odamlar boshqa prototiplarga ega bo'lishi mumkin. Misol uchun, "meva" tushunchasi uchun ko'pincha "apelsin" prototipi deb ataladi. Bu prototipdan kontseptsiya vakili sifatida foydalanishga fundamental cheklovlar qo'ymaydi. Shunday qilib, tushunchalarni ularning tipik vakillari - prototiplari yordamida tavsiflash mumkin va bu ob'ektning tushuncha prototipidan semantik masofa o'lchovi yordamida ma'lum bir ob'ektning berilgan tushunchaga tegishli yoki tegishli emasligini aniqlash mumkin. Faqatgina kiritilishi mumkin bo'lgan tushunchalar mavjud. Tushunchalarni aniqlashning turli usullari bilan bog'liq masalalar 2.1-bo'limda batafsil ko'rib chiqiladi. Kontseptualizatsiya har doim dunyoning soddalashtirilgan ko'rinishidir. Mavzu sohasidan faqat berilgan topshiriq4 uchun zarur bo'lgan elementlar tanlanadi, fan sohasining boshqa xususiyatlari hisobga olinmaydi. Keling, darslikka misol keltiraylik. Muayyan uy ob'ekt sifatida harakat qilsin. Uyni qurishda bir nechta mutaxassislar, jumladan me'morlar, materiallar yetkazib beruvchilar va quruvchilar ishtirok etmoqda. Arxitektorlar uyni bir butun sifatida ko'rib, asosan turli qismlar (xonalar, pollar) bu butunlikni qanday tashkil etishiga e'tibor berishadi. Quruvchilar ham uyni yaxlit deb hisoblashadi, lekin asosiy e'tibor ishlab chiqarish jarayonlariga qaratiladi: polni qurish uchun qancha odam va qanday jihozlarni jo'natish, qurish uchun qancha vaqt ketadi va hokazo. Material yetkazib beruvchilar uyni umuman ko'rmaydilar.

Yetkazib beruvchining fikriga ko'ra, uy bir nechta tonna tsement, beton, po'lat va boshqa qurilish materiallaridan iborat idishdir.

Shunday qilib, maqsad kontseptualizatsiya qanday amalga oshirilishini belgilaydi. "Kontseptualizatsiya" atamasi Gruber tomonidan ontologiyani qurish jarayoniga ishora qilish uchun ham ishlatilgan. Kontseptuallashtirish natijasini kontseptual sxema deb ataymiz. Ushbu atama taxminan bir xil ma'noga ega bo'lgan ma'lumotlar bazalarida qo'llaniladi, lekin ba'zi ishlab chiqarish vazifalari uchun ma'lumotlar bazasi sxemasini aniqlashning maxsus holatida. Terminologiyada ham bir oz nomuvofiqlik mavjud. Yuqorida biz ontologiyani muammoli sohani kontseptsiyalashning spetsifikatsiyasi sifatida aniqladik. Ammo, falsafada "ontologiya" atamasi haqiqatni tushunchalar va ular orasidagi aloqalar to'plami shaklida tavsiflashni anglatadi, ya'ni mohiyatan voqelikning kontseptualizatsiyasi. Muhandislikda muhandislik muammosining predmeti kontseptuallashtiriladi va bu kontseptsiyaning maqsadi odatda sof utilitar bo'lib, hal qilinayotgan muhandislik muammosining mazmuni bilan bog'liq. Shunday qilib, maqsadlarni aniq ajratish mavjud:

- Falsafada ontologiya voqelik xususiyatlarini o'rganish uchun yaratilgan.
- Muhandislikdagi ontologiya muhandislik muammosini hal qilish uchun yaratilgan.

Kontseptuallashtirish maqsadlari uchun bu nomuvofiqlik quyidagi taklif uchun turtki bo'ldi: ontologiya atamasi faqat bunday kontseptualizatsiyani yaratish uchun maxsus yaratilgan voqelikni kontseptsiyalashning spetsifikatsiyalariga ishora qilish uchun ishlatiladi va kontseptualizatsiyaning barcha boshqa spetsifikatsiyalarini belgilash uchun ishlatiladi. kontseptual model atamasi bilan. Shunga ko'ra, ontologiya - bu kontseptsiyalarning rasmiy belgilangan sxemasi ko'rinishidagi voqelikning tavsifi va kontseptual sxema bir xil, ammo faqat ba'zi muhandislik muammolarining mavzu sohasiga nisbatan. Muallifga bunday bo'linish ortiqcha bo'lib tuyuladi, lekin, albatta, o'quvchi ushbu terminologiyadan foydalanishga haqli. Modellashtirish atamasi kontseptual diagrammani qurish jarayoniga nisbatan ham qo'llaniladi. Yuqorida aytib o'tilganidek, muhandislik ontologiyasi sohasidagi ko'plab mutaxassislar ontologik modellashtirish va kontseptual modellashtirishni farqlaydilar.

Bu yerda kontseptual sxema va muhandislik ontologiyasining aniq ta'riflarini keltirib o'tamiz.

Ta'rif 1.2. *Muhandislik topshirig'ining kontseptual diagrammasi - bu topshiriqning predmeti sohasining mavhum soddalashtirilgan ko'rinishi, u faqat ushbu vazifani bajarish uchun foydali bo'lgan mavzu sohasining xususiyatlarini tanlashda ifodalanadi va uning qolgan xususiyatlarini hisobga olishdan voz kechish.*

Ta'rif 1.3. *Muhandislik ontologiyasi - muhandislik muammosining kontseptual loyihasining rasmiy tavsifi (spetsifikatsiyasi).*

1.2. Ontologiya turlari

Ushbu bo'limda biz eng oddiy taksonomiyalardan tortib, ularning mazmunidagi o'zgarishlarni qo'llab-quvvatlovchi murakkab ontologiyalargacha bo'lgan har xil turdagi ontologiyalarni ko'rib chiqamiz.

Tasniflar

Tasniflash kontseptual sxema spetsifikatsiyasining oddiy shakli bo'lib, unda elementning barcha elementlari turlarga bo'linadi va kiritilgan turlar o'rtasida ierarxik munosabatlar kiritiladi. Agar berilgan tur tomonidan ko'rsatilgan sinfda kichik tip sinfining barcha namunalari mavjud bo'lsa, bu tur boshqasining super turi hisoblanadi. Ierarxiya munosabatlarining quyi to'plam munosabatlari orqali ta'rifidan kelib chiqadiki, ierarxiya munosabatlari o'tishli, ya'ni, agar tip turning pastki turi bo'lsa va tip - turning pastki turi bo'lsa, u holda *A* turi - turning kichik turi. Tasniflash Aristotel an'analariga asoslanib, tushunchalarni xususiyatlar to'plami sifatida belgilaydi.

Tabiiy dunyo

Shohlik

Sinf

Ajralish

Jins

Ko'rinish

Turli xillik

Jadval 1.1. *Linneevski ierarxiyasi elementlari.*

Eng mashhur tasniflash tizimlaridan biri shved tabiatshunosi Karl Linney nomi bilan atalgan Linnean ierarxiyasidir. Linnean ierarxiyasi - tabiat dunyosining ilmiy tasnifi. Ildiz turi "tabiiy dunyo"

bo'lib, u dastlab Linney tomonidan shohliklar deb ataladigan uchta kichik turga bo'lingan: "mineral", "sabzavot" va "hayvon". Endi, biologik tasnifda shohliklar "tabiiy dunyo" turining sakkizta kichik turi bo'lib, biz ular haqida batafsil to'xtalmaymiz. Shohlik tipining bevosita kichik turi sinf tipi edi. Dastlab, Linneya ierarxiyasida faqat oltita sinf mavjud edi: sutemizuvchilar, qushlar, sudraluvchilar, baliqlar, hasharotlar va qurtlar. Sinfning kenja turi "otryad", uning kichik turi "tur", turkumning kichik turi "tur" va "tabiat dunyosi" predmet sohasining elementi "tur" deb atalgan. Ierarxiya darajasi daraja deb ataladi, shuning uchun Linnean tasnifida olti daraja mavjud bo'lib, 1.1-jadvalda darajalarning kamayish tartibida keltirilgan.

Masalan, Pyotr Ivanovning "tabiiy dunyo" (turli) predmeti elementi quyidagicha tasniflanadi:

"hayvonot" olami, "sutemizuvchilar" sinfi, "primatlar" otryadi, "odam" jinsi, "inson aqli" turlari. Endi biologik tasnif sezilarli darajada o'zgardi, yangi darajalar paydo bo'ldi va har bir darajaning belgilari (xususiyatlari) soni sezilarli darajada kengaydi, lekin tasniflash tamoyillari o'zgarishsiz qolmoqda.

Tur ierarxiyasi bilan tasniflash uchun ko'pincha taksonomiya atamasi qo'llaniladi. Taksonomiya - bu ierarxiya munosabatlari turlar daraxti sifatida ifodalanishi mumkin bo'lgan tasnifdir, ya'ni taksonomiyada har qanday tur bir nechta supertipga ega bo'lishi mumkin emas. Yangi "yengil ontologiya" atamasi ko'pincha muhandislik ontologiyalari bo'yicha mutaxassislar tomonidan tasniflash uchun sinonim sifatida ishlatiladi. Bu atama, shubhasiz, "og'ir ontologiyalar" ham mavjudligini anglatadi. Oxirgi atama elementlar orasidagi turlarning ierarxiyasini ta'kidlashdan tashqari, odatda, maxsus munosabatlar va funktsiyalar bilan ifodalangan yanada murakkab munosabatlar mavjud bo'lgan ontologiyalarni bildiradi. Modellashtirish uchun og'ir ontologiyalardan foydalanishni talab qiladigan muammolarning katta sinfi mavjud. Shu ma'noda, og'ir ontologiyalarni modellashtirish odatiy hol emas. Og'ir ontologiyalarning oddiy misollari quyida "Aloqalar bilan ontologiyalar" bo'limida keltirilgan. Qizig'i shundaki, Linnean tasnifining o'xshashligi dasturlash metodologiyasida kutilmaganda o'zini namoyon qildi. Bugungi kunda eng zamonaviy dasturlash paradigmalaridan biri ob'ektga yo'naltirilgan dasturlash (OOP) deb ataladi. OOP sinflar misollari bo'lgan ob'ektlarda ishlaydi. OOP klassi

noyob nomga ega bo'lgan atributlar va usullar to'plamidir. Sinf atributining turi bor va sinfning ma'lum bir namunasi uning atributlarini ma'lum qiymatlar bilan to'ldiradi. Sinf usuli - bu ma'lum bir sinf misolida chaqirilishi mumkin bo'lgan algoritm bo'lib, usulga o'tkazilgan ushbu sinf misolining qiymatlari va atribut qiymatlari asosida muayyan parametrlar to'plamini oladi. Siz meros mexanizmi deb ataladigan narsadan foydalanib, sinflar o'rtasida ierarxik munosabatlarni o'rnatishingiz mumkin. Sinf tipini e'lon qilishda uning bir yoki bir nechta ota-ona turlaridan "meros olishi" ko'rsatiladi, ya'ni ota-onalarning barcha atributlari va usullari endi bolalar sinfining atributlari va usullari hisoblanadi.

Shubhasiz, bunday dasturlash tili metodologiyasi dasturning kontseptual sxemasining tasnifi ko'rinishidagi tavsifini belgilaydi, bu dasturlash jarayonini, ayniqsa katta dasturlar uchun soddalashtiradi, chunki dastur elementlarini aniq belgilangan interfeys va ular o'rtasida ma'lumotlar almashinuvi uchun aniq belgilangan protokolga ega modellarga bo'lish imkonini beradi. OOP nazariyasi 70-yillarning oxirida asos solingan va Barbara Liskov tomonidan kashshof bo'lgan. Uning yondashuvi Linnaean tasnifi bilan o'xshashlikka asoslangan va terminologiya undan meros bo'lib qolgan.

Shubhasiz, bunday dasturlash tili metodologiyasi dasturning kontseptual sxemasining tasnifi ko'rinishidagi tavsifini belgilaydi, bu dasturlash jarayonini, ayniqsa katta dasturlar uchun soddalashtiradi, chunki dastur elementlarini aniq belgilangan interfeys va ular o'rtasida ma'lumotlar almashinuvi uchun aniq belgilangan protokolga ega modellarga bo'lish imkonini beradi. OOP nazariyasiga 70-yillarning oxirida asos solingan va Barbara Liskov tomonidan taqdim etilgan. Uning yondashuvi Linnaean tasnifi bilan o'xshashlikka asoslangan va terminologiya undan meros bo'lib qolgan.

Tasniflashning yana bir mashhur namunasi D.I. kimyoviy elementlarning davriy jadvalidir. Ushbu tasnifdagi ildiz turi "kimyoviy element" dir. Kimyoviy element bir nechta atributlarga ega, ularning asosiysi atom og'irligi. Kimyoviy elementning qolgan xususiyatlari uning atom og'irligi bilan belgilanadi. Davriy jadval taksonomiya emas, chunki uning tasnifi ierarxik emas, balki eng muhim "davr" tushunchasiga asoslanadi. Mendeleyev kashfiyotining mohiyati shundan iboratki, kimyoviy elementlarning atom og'irligi ortishi bilan ularning xossalari monoton emas, balki davriy ravishda o'zgaradi.

Atom og'irligi ortib borayotgan turli xil xususiyatlarga ega elementlarning ma'lum sonidan keyin xususiyatlar takrorlana boshlaydi. Masalan, natriy kaliyga, neon argonga, oltin kumush va misga o'xshaydi. Albatta, xususiyatlar aniq takrorlanmaydi, asosiy yoki ular aytganidek, ma'lum bir davr uchun muhim bo'lgan xususiyatlar o'zgarishsiz qoladi. Shunday qilib, kimyoviy elementlarni davrga mansubligi bo'yicha tasniflash mumkin. Bundan tashqari, davrdagi kimyoviy elementning tartib raqami ham muhim: davrda bir xil tartib raqamiga ega bo'lgan barcha elementlar o'xshash xususiyatlarga ega. Shunday qilib, neon va argon o'z davrlarining oxirgi elementlarini ifodalaydi va inert gazlar oilasiga kiradi. Xuddi shu kimyoviy element, masalan, neon, birinchi davrga tegishli, ya'ni "birinchi davr" turi, balki "inert gazlar" turiga ham kiradi. Demak, davriy sistemaning tasnifi taksonomiya emas. Bu tasniflashning murakkabroq turiga misol bo'lib, barcha tasniflar taksonomiya emasligini ko'rsatadi.

Web-qidiruv tizimlari endi ierarxik jihatdan bog'liq bo'lgan qidiruv toifalarini tavsiflovchi web-saytlar tasnifidan ko'p foydalanadi. Odatda, bunday tasnif kengaytirilgan taksonomiya bo'lib, unda har bir element qo'shimcha xususiyatlarga ega bo'lishi mumkin bo'lgan qidiruv toifalari daraxti mavjud. Ushbu tasnif, aslida, qidiruv vazifalarini optimallashtirish maqsadida qurilgan internetning taksonomiyasidir. Misol uchun, Yandex kompaniyasining tasnifi Yandex katalogi deb ataladi va to'rtta ierarxiya darajasidan ko'p bo'lmagan daraxtdir.

Ushbu tasnifdagi turlar "sarlavhalar" deb ataladi. Quyidagi yuqori darajadagi sarlavhalar mavjud:

- *Yuqori texnologiyalar*
- *Ishlash*
- *O'qish*
- *Uy*
- *Jamiyat*
- *O'yin-kulgi*
- *Dam olish*
- *Madaniyat*
- *Sport*
- *Ommaviy axborot vositalari*
- *Ishlab chiqarish*

- *Biznes*
- *Yordam*
- *Avtomatik*
- *Portallar*

Shunga o'xshash tasniflar barcha mashhur qidiruv tizimlarida mavjud.

Nisbatli ontologiyalar

Klassifikatsiyalar ontologiyaning eng keng tarqalgan turi bo'lsada, kontseptual sxemalari oddiy tasniflash doirasiga to'g'ri kelmaydigan muammolar mavjud. Keling, oddiy misolni ko'rib chiqaylik. Geometrik figuralar ontologiyasi proyeksiyalansin va bu ontologiyada “kvadrat” turini tavsiflash zarur. Ushbu turni ikki shaklda tavsiflash mumkin:

1. Tomonlari bir-biriga teng bo'lgan to'rtburchaklar kabi.
2. Burchaklari teng bo'lgan romb kabi.

Ikkala holatda ham oddiy tasnif yetarli emas, qandaydir tarzda to'rtburchaklar tomonlari yoki rombnings burchaklari o'rtasidagi munosabatni o'rnatish kerak. Umuman olganda, turlarni atributlarning oddiy ro'yxati bilan tavsiflash juda qiyin. Misol uchun, yengil avtomobilni og'irligi uch yarim tonnadan kam bo'lgan avtomobil deb ta'riflash mumkin. Boshqacha qilib aytganda, yengil avtomobil uning atributining qiymatini cheklash orqali ko'rsatiladi. Bu munosabatlarning eng ko'p qo'llaniladigan turi. Bundan tashqari, murakkabroq holatlar mavjud. Masalan, mexanik harakat ontologiyasidagi “yo'l”, “vaqt” va “tezlik” tushunchalari o'zaro bog'langan bo'lib, faqat ma'lum funksiya yordamida ifodalanadi. Bunday vazifalarni bajarish uchun murakkabroq bog'liqliklarni ifodalovchi ontologiyalarni qurish imkoniyatiga ega bo'lish kerak. Va, albatta, buning uchun ushbu ifoda imkoniyatlarini o'z ichiga olgan maxsus til kerak. Odatda, bunday til sifatida u yoki bu turdagi predikativ hisoblar tili qo'llaniladi, ya'ni belgilangan ontologiya mantiqiy nazariyadir. Misol tariqasida, Ontolingua tizimida ontologiyalarni aniqlash uchun til sifatida qo'llaniladigan Knowledge Interchange Format (KIF) tilini ko'rib chiqaylik. Keyinchalik (2-bob) biz ushbu tilni batafsil o'rganamiz, bu yerda biz uning ba'zi g'oyalarini ko'rib chiqamiz, ularni tushunish misollarni o'zlashtirish uchun zarurdir.

KIF tili aslida funksiya belgilari va tenglikni o'z ichiga olgan predikat hisobidir. KIF LISP tiliga asoslangan va shuning uchun

prefiks belgisi mavjud. KIF predmet sohasini uchta asosiy tushunchaga asoslanadi: to'plam, munosabat va funktsiya. To'plam - bu ob'ektning mavhumligi, ya'ni: predmet sohasi dunyodagi barcha ob'ektlar to'plamidir, deb ishoniladi, aslida mavjud yoki farazdir. KIF tilidagi predmet sohasi abstraksiya bo'lganligi sababli, har xil foydalanuvchilarga o'z predmet sohaslariga ega bo'lishiga ruxsat beriladi, biroq bunday barcha predmet sohalari quyidagi asosiy ob'yektlarni o'z ichiga olishi kerak:

1. So'zlar. Boshqacha qilib aytganda, so'zlar har qanday predmet sohasining ob'ekti bo'lib, uning konseptual sxemasi KIF tili bilan modellashtirilgan.
2. Barcha kompleks raqamlar.
3. Element ob'ektlarining barcha cheklangan ro'yxatlari.
4. Elementdagi barcha ob'ektlar to'plami.
5. \perp bilan belgilangan maxsus ob'ekt, bu funktsiya aniqlanmagan argumentlar bilan chaqirilgan har qanday funktsiyaning qiymati hisoblanadi.

Funktsiya - bu ob'ektlar o'rtasidagi munosabatlarning bir turi. Argumentlar deb ataladigan ob'ektlarning har bir chekli ketma-ketligi bilan funktsiya funktsiya qiymati deb ataladigan ba'zi ob'ektlarni bog'laydi. Rasmiy ravishda, funktsiya - bu funktsiyaning mumkin bo'lgan argumentlarini ifodalovchi ob'ektlarning cheklangan ro'yxatlari to'plami va har bir ro'yxatning oxirgi elementi berilgan argumentlar to'plami uchun funktsiyaning qiymatidir. Masalan, “2 **” funksiyasi $\langle 4, 8 \rangle$ yoki $\langle 17, 34 \rangle$ ko'rinishdagi ikki elementli ro'yxatlar to'plami bo'lib, bunda ro'yxatning birinchi elementi funktsiya argumenti, ikkinchi elementi esa uning qiymati hisoblanadi. . Aloqa - bu elementdagi ob'ektlar o'rtasidagi munosabatlarning yana bir turi. Aloqa - bu element ob'ektlarining chekli ro'yxatlarining ixtiyoriy to'plami va bu ro'yxatlar umumiy holatda o'zgaruvchan uzunlikda bo'lishi mumkin. Har bir ro'yxatning elementlari berilgan munosabatni qanoatlantirishi yoki munosabat shu elementlarda bajarilganligi aytiladi. Masalan, “ \langle ” munosabati $\langle 2, 3 \rangle$ va $\langle 45, 102 \rangle$ ro'yxatlarini o'z ichiga oladi, lekin $\langle 5, 2 \rangle$ ro'yxatini o'z ichiga olmaydi.

KIF tili munosabatlar va funktsiyalardan foydalangan holda element ontologiyasini tavsiflaydi. Ushbu asosiy tushunchalar yordamida turlarni ham tavsiflash mumkin. Turlar maxsus element

obyektlaridir. Ob'ektlar orasidagi turlarni ajratish uchun "Sinf" unar munosabati qo'llaniladi. Agar berilgan ob'ekt "Sinf" munosabatini qanoatlantirsa, u holda u sinf hisoblanadi. Biroq, shu bilan birga, har bir sinfning o'zi bir o'rinli munosabatni ifodalaydi, bu uning ko'plab misollarini ajratib turadi. Aloqalar mantiqiy operatsiyalar yordamida bir-biri bilan bog'lanishi mumkin. Bular odatiy mantiqiy bog'lovchilar "yoki" (yoki), "va" (va) va inkor (yo'q), shuningdek, mantiqiy ketma-ketlikning amali (implikasiya, ya'ni A rost bo'lsa, B ham to'g'ri) bilan belgilanadi. KIF tilidagi operatorlarda " \Rightarrow " ($A \Rightarrow B$), shuningdek, uning ekvivalentlari " $= <$ " (qarama-qarshi yo'nalishda, ya'ni $A = < B \Rightarrow A$ bilan bir xil), " \Leftarrow " (har ikki yo'nalishda ham, ya'ni $A \Leftarrow B \Rightarrow B$ va $B \Rightarrow A$ ning bir vaqtda bajarilishiga teng). Tenglik ham mavjud. O'zgaruvchilar "?" bilan boshlanadi. Biz KIF tilini tizim tavsifiga bag'ishlangan 4-bobda batafsil bayon qilamiz. Ontolingua, bu yerda biz o'quvchining qiziqishini qondiramiz va boshlaymiz aniq misollarni muhokama qilish.

1.1-misol. (Qisman buyurtma)

Qisman buyurtma berish amaliyotda keng tarqalgan. Odatiy holat shundaki, ba'zi ob'ektlar bir-biri bilan taqqoslansa, boshqalari esa yo'q. Bundan tashqari, har bir ob'ektni o'zi bilan solishtirish mumkin. Bu xususiyat reflekslik deb ataladi. Qisman tartib munosabatining boshqa ikkita xususiyati - antisimetriya va tranzitivlik - quyida rasman tavsiflangan. Qisman tartib munosabatini "6" bilan belgilaymiz. Matematik jihatdan uning uchta xususiyatini quyidagicha ifodalash mumkin:

1. Reflektorlik. Har qanday ob'ekt uchun a , $a \ 6 \ a$ bajariladi.
2. Antisimetriya. Agar $a \ 6 \ b$ va $b \ 6 \ a$ bo'lsa, $a = b$.
3. Tranzitivlik. Agar $a \ 6 \ b$ va $b \ 6 \ c$ bo'lsa, $a \ 6 \ c$ bajariladi.

Demak, qisman tartib bu uchta xususiyatga ega bo'lgan har qanday ikki o'rinli munosabatdir. Avval refleksiv munosabatni, antisimetrik munosabatni va o'tish munosabatini aniqlash mumkin, keyin esa qisman tartib munosabatini ham refleksiv, ham antisimetrik va o'tish munosabati sifatida belgilash mumkin.

Ontolingua:

(defrelation partial-order
 \Leftarrow (partial-order ?r ?s)
 (and (transitive ?r ?s)

(asymmetric ?r ?s)
(reflexive ?r ?s))))

Lisp sintaksisida $f(x, y)$ ko‘rinishdagi funksiya ifodalari ro‘yxat $(f \ x \ y)$ ko‘rinishida ifodalanadi, bunda birinchi element funksiya nomi, qolgan ro‘yxat elementlari esa uning argumentlarini ifodalaydi. Ro‘yxat elementlari bo‘sh joylar bilan ajratilgan. O‘zaro munosabatlarni aniqlash uchun "defrelation" kalit so‘zi ishlatiladi. "? S" to‘plamidagi "? R" o‘zgaruvchisi bilan belgilanadigan qisman tartibli munosabat aniqlanadi. "? R" munosabati "? S" to‘plamidagi qisman tartibdir, agar u bir vaqtning o‘zida "? S" to‘plamida o‘tish, antisimmetrik va refleksli bo‘lsa. Albatta, oxirgi uchta munosabatlar hali ham aniqlanishi kerak.

Masalan, refleksiv munosabatning ta‘rifi:

(defrelation reflexive
(=<=> (reflexive ?r, ?s)
(and (binrel ?r, ?s)
(forall ?x)
(=> (member ?x, ?s)
(holds ?r, ?x, ?x))))

Yuqoridagidek, "? S" to‘plamida aniqlanmagan "? R" munosabati, agar ikkita shart bir vaqtning o‘zida bajarilgan bo‘lsa, refleksli hisoblanadi:

1. "? R" ikkilik munosabat (binrel? R, ? S), ya‘ni har doim ikkita argumentga ega.
2. "? S" (a‘zo? X, ? S) to‘plamining har bir "? X" (forall? X) ob‘ekti uchun quyidagilar bajariladi: "? X" o‘zlari bilan taqqoslanadi (tutadi? R, ? X, ? X).

Ehtimol, bu yerda izohga muhtoj bo‘lgan yagona narsa argumentlar ro‘yxati birinchi parametr sifatida berilgan munosabatda yoki yo‘qligini tekshirish uchun ishlatiladigan holds munosabatidir.

O‘zgaruvchan ontologiyalar

Hozircha ontologiyalar vaqt o‘tishi bilan o‘zgarishi mumkinligiga e‘tibor bermaganmiz. Yuqorida keltirilgan ontologiya misollari, go‘yo kontseptual sxemaning oniy tavsiflari bo‘lib, uning vaqt o‘zgarishini aks ettirmaydi. Shu bilan birga, kontseptual sxemalar, qoida tariqasida, o‘zgaradi, yangi tushunchalar paydo bo‘ladi va ba‘zi eskilari yo‘qoladi, terminologiya o‘zgaradi, vaqt o‘tishi bilan muhim va ahamiyatli hisoblangan narsalar vaqt o‘tishi bilan o‘z

ahamiyatini yo'qotadi, lekin ilgari hech kim e'tibor bermagan narsa birdaniga o'zgaradi. alohida ma'noga ega va odamlarning e'tiborini tortadi. Ushbu o'zgarishlarni tabiiy tarzda ko'rsatish uchun kontseptual sxemani o'zgartirishning ushbu imkoniyatlarini uning ontologiyasi qurilgan modellashtirish vositasiga qo'yish yaxshi bo'lar edi. Albatta, ontologiyaning o'zida bunday imkoniyatlarni ko'rsatishning hojati bo'lmaganda muammolar mavjud, sodir bo'lgan o'zgarishlarni aks ettiruvchi ontologiyaning yangi versiyasini chiqarish osonroq. Shunday qilib, ehtimol, yuqorida ko'rib chiqilgan tasniflar uchun uni ontologiyaga o'zgartirish imkoniyatlarini kiritish ortiqcha bo'lib ko'rinadi, faqat asosiy ob'ektlarni tushunishni qiyinlashtiradi.

Ammo haqiqatga yaqinroq bo'lgan boshqa ko'plab vazifalar uchun, shubhasiz, o'zgarishlarni qo'llab-quvvatlash vositalari zarur. Biz hozirgacha e'tibordan chetda qoldirgan yana bir haqiqat, ontologiyalar ijtimoiy mahsulotdir, ya'ni jamoaviy mehnat natijasi. Ontologiyani bir kishi yaratishi mumkin, lekin u faqat boshqa odamlar tomonidan qo'llanilganda foydali bo'ladi. Demak, jamiyat a'zolari o'rtasida hozirda qurilayotgan kontseptual sxemaning mazmuni bo'yicha kelishuv mavjud. Tushunishning asosiy qurilish bloklari sifatida tushunchalar haqida gaplashdik. Ammo tushunchalar jamiyat a'zolari o'rtasida ularni talqin qilish bo'yicha kelishuvga erishilgandagina shunday bo'ladi. Shu vaqtgacha tushunchalar faqat individual kechinmalar (ma'nolar) bo'ladi. Tushunchaga ma'no berish munozara orqali sodir bo'ladi, uning davomida tushuncha oydinlashadi va aniq shaklga ega bo'ladi. Ushbu munozaraning natijasi jamoa a'zolari o'rtasida o'zlarining shaxsiy tajribalarini bir-birlari bilan qanday bog'lashlari haqida kelishuvdir. Ontologiyalarni yaratish uchun dasturiy ta'minot tizimi ularni jamoaviy tahrir qilish imkoniyatini yaratsa yaxshi bo'lardi. Bunday imkoniyatlarni amalga oshirishning o'ziga xos qiyinchiliklari va kamchiliklari bor.

Turlarni o'zgartirish jarayonida ularning hajmi ham o'zgarishi mumkin, ya'ni yangi misollar paydo bo'lishi mumkin, boshqa misollar yo'qolishi mumkin. Shu nuqtai nazardan qaraganda, to'plam yordamida turni modellashtirish ancha noqulay, chunki to'plamlar ularning elementlari bilan belgilanadi va ikkinchisi hali hammasiga ma'lum bo'lmasligi mumkin. Turning to'liq bo'lmagan tavsifi xususiyatini yetarli darajada ifodalash uchun tegishli rasmiy apparat talab qilinadi. Bunday rasmiy apparat kategoriyalar nazariyasidir.

Kategoriya nazariyasi matematikaning "kategoriya" deb nomlangan matematik tuzilmalar bilan ishlaydigan bo'limidir. Turkum ikki turdagi elementlarni o'z ichiga oladi: ob'ektlar va morfizmlar. Morfizmlar ob'ektlar orasidagi bog'lanishni ifodalaydi. Shunday qilib, to'plamlar toifasi ob'ektlar sifatida to'plamlarni va morfizm sifatida to'plamlarning bir-biriga moslashishini o'z ichiga oladi. Har bir morfizm ushbu turkumdagi ikkita ob'ektga mos keladi: morfizmning boshlanishi bilan bog'liq ob'ekt va morfizmning oxiri bilan bog'liq bo'lgan ob'ekt. Ob'ektlar to'plam bo'lgan va morfizmlar to'plam xaritalari bo'lgan to'plamlar toifasi uchun morfizmning boshlanishi bilan bog'liq ob'ekt tegishli ravishda berilgan morfizm xaritalash sohasi va morfizmning oxiri bilan bog'liq ob'ekt hisoblanadi. qiymatlar diapazoni hisoblanadi. An'anaga ko'ra, morfizmlar lotin alifbosining kichik harflari bilan, ob'ektlar esa lotin alifbosining bosh harflari bilan belgilanadi (notatsiya, to'plamlar va ular orasidagi funktsiyalarning an'anaviy belgilaridan meros). Biz f yozamiz: $A \rightarrow B$ morfizmi f ob'ektdan boshlanib, B ob'ektda tugaydigan morfizmi belgilash uchun. Morfizmlar o'rtasida morfizmlarni tuzish yoki ko'paytirish operatsiyasi bo'lishi kerak, xususan: agar $f: A \rightarrow B$ va $g: B \rightarrow C$ bo'lsa, u holda $\square = fg$ morfizmi $\square = fgs$ morfizmi mavjud: , bu yerda birinchi morfizmning oxiri ikkinchisining boshiga to'g'ri keladi (morfizmlarning tarkibi f va g fg yoki $f \circ g$ sifatida belgilanadi). Shubhasiz, to'plamlar kategoriyasi misolimizda morfizmlar tarkibi funktsiyalar tarkibiga to'g'ri keladi.

Ob'ektlar orasidagi morfizmlar ma'lum talablarga javob berishi kerak, xususan:

- Har bir ob'ekt uchun A , bir xil morfizm $A: A \rightarrow \rightarrow \rightarrow$ va $\rightarrow \rightarrow g \rightarrow \uparrow \circ \circ \circ idA g g g g \circ \circ \circ \circ$ Boshqacha qilib aytganda, bir xil morfizmning boshqa har qanday morfizm bilan tarkibi bir xil kerakli morfizmga olib keladi. Identifikatsiya xaritasi har qanday to'plamda aynan shunday ishlaydi (ya'ni, to'plamning har bir elementini o'ziga oladigan xaritalash).

- Morfizmlar tarkibi assotsiativlik xususiyatiga ega: agar morfizmlar $f: A \rightarrow B$, $g: B \rightarrow C$ va $\square: C \rightarrow D$ morfizmlari berilsa, $f \rightarrow D$ morfizmlari berilgan bo'lsa, $f(D)$ bo'lishi shart. Boshqacha qilib aytganda, uchta (va shuning uchun har qanday raqam) morfizmning tarkibi har qanday tartibda olinishi mumkin, natijada biz bir xil o'qni

olamiz. Ma'lum bo'lishicha, har qanday to'plam-nazariy konstruksiyani kategoriyalar nazariyasi vositalari yordamida aniqlash mumkin, ya'ni berilgan konstruksiya amal qiladigan ob'ektlar to'plami va bu ob'ektlar orasidagi morfizmlar to'plami orqali. Bunday tasavvurlar juda kutilmagan va ko'pincha ma'lum bir matematik konstruksiyani aniqlashtirish uchun foydalidir. Kategoriya tushunchasi bu matematik formalizm nomini belgilab bergan har qanday to'plam-nazariy konstruksiya asosida yotadi – kategoriya xuddi shu faylasuflarning ontologik nazariyalari uchun Kant va Aristotelning falsafiy kategoriyalari kabi matematik konstruksiyalar uchun asos bo'lib xizmat qiladi. Kategoriyalar nazariyasida to'plamlar ustidagi turli amallarning analoglarini aniqlash mumkin, masalan, to'plamlarning dekart mahsuloti yoki ularning kesishuvlari. Dekart mahsuloti $X \times Y$ ham to'plam, bu holda juftlar to'plami va shuning uchun to'plamlar toifasi ob'ektidir. Har qanday ikkita to'plam uchun ularning Dekart mahsuloti mavjud, shuning uchun Dekart mahsuloti toifadagi har qanday ikkita ob'ektni uchinchi ob'ekt bilan taqqoslaydigan maxsus operatsiya sifatida ko'rsatilishi mumkin. Bunday operatsiyalar universal konstruksiyalar yoki chegaralar deb ataladi. To'plamlar uchun siz ularning Dekart ko'paytmasi, yig'indisi, ko'rsatkichi va boshqa amallarini belgilashingiz mumkin. Oddiy to'plamlar uchun belgilangan ob'ektlar uchun barcha amallar aniqlangan toifaga topos deyiladi. Shunday qilib, topos "barcha to'plamlar to'plami" yoki matematiklar aytganidek, "koinot" tushunchasining qat'iy ifodasidir. Topos ob'ektlari ularning mazmuniga ko'ra (to'plamlardagi kabi) emas, balki ularning boshqa ob'ektlar bilan aloqasi bilan belgilanadi. Topos ontologiyalarni modellashtirish uchun juda mos keladi, ularda tushunchalar to'liq bo'lmagan turlar yordamida aniqlanadi. Tushunchalar ba'zi bir topos ob'ektlari sifatida, tushunchalar orasidagi turli munosabatlar esa morfizm shaklida ifodalanishi mumkin. Universal konstruksiyalar yordamida yangi tushunchalarni shakllantirish mumkin. Masalan, toposda ikkita maxsus obyekt A va B bor deylik, birinchisi "erkak" tushunchasini, ikkinchisi esa "ayol" tushunchasini modellashtiradi. Keyin "mumkin bo'lgan er-xotinlar" tushunchasini bildiruvchi ob'ektni qurish mumkin, bu ob'ekt $A \times B$ bilan "sub'ekt munosabatlari" deb nomlangan maxsus munosabatda bo'ladi. Sub-ob'ekt munosabati - bu

bir to'plam boshqa to'planning kichik to'plami ekanligining kategorik analogidir. "Haqiqiy turmush qurgan juftliklar" kontseptsiyasini joriy qilish uchun buning uchun mumkin bo'lgan juftliklardan faqat haqiqatda mavjud bo'lganlarni tanlaydigan maxsus ob'ektni qurish kerak. Buning uchun ob'ektning elementlari - maxsus cheklangan ob'ektning morfizmlari, bir elementli to'planning analogidan foydalanish kerak.

Ba'zan shunday bo'ladiki, hatto toposning tuzilishi ham yangi tushunchalarni qurish jarayonlarini to'liq aks ettirmaydi. Masalan, kontseptual aralashtirish usuli bilan yangi tushunchalarni shakllantirish qurilishini ko'rib chiqing. Kontseptual aralashtirish ko'pincha assimetrikdir. Masalan, ikkita tushuncha mavjud bo'lsa: "qayiq" va "uy" bo'lsa, "uy-qayiq" tushunchasi "qayiq-uy" tushunchasi bilan mutlaqo bir xil emas. Limit konstruktsiyasi yordamida bunday assimetrik aralashtirishni ifodalash qiyin, chunki chegaralar ular asosida qurilgan ob'ektlar va morfizmlarga nisbatan simmetrikdir. Shu sababli, tushunchalarning ta'riflarida asosiy (majburiy) xususiyatlarni ajratib ko'rsatish va ushbu tanlovni hisobga olgan holda tushunchalarni aralashtirish operatsiyalarini o'zgartirish qobiliyatini kiritish kerak. Shunga ko'ra, aralashtirish tushunchalarining ishlashini simulyatsiya qiluvchi matematik konstruktsiyalarning ta'riflarini o'zgartirish kerak. Bunday urinishlar allaqachon Jozef Gogen tomonidan amalga oshirilgan, buning natijasida 2/3-toifalar deb ataladigan narsa paydo bo'lgan. Kategoriyalar nazariyasi formalizmiga asoslangan ontologik modellashtirish alohida katta mavzudir. Ushbu masalalar ushbu monografiyada batafsil muhokama qilinmaydi.

Monografiyada nima yozilgan

Monografiya muhandislik bilimlarining turli sohalarida ontologiyalarni qurish uchun zamonaviy tizimlar taqdimotiga bag'ishlangan. Ontologik modellashtirishga zamonaviy yondashuvlar, asosan, ikkinchi darajali tillarning ba'zi xususiyatlari qo'shilgan birinchi darajali predikatlar hisobi tilida kontseptual sxemani ifodalashga asoslanadi: munosabatlarning xususiyatlariga turli xil cheklovlar (predikatlar). Shunday qilib, monografiyada keltirilgan eng mashhur ontologiya boshqaruv tizimlarining tavsiflari, u yoki bu tarzda, predikat hisobiga tegishli. Ta'riflangan ontologiyani boshqarish tizimlarini taqdim etishda predikatlar hisobi tillari bilan bog'liq turli jihatlarni eslatib o'tish zarur deb hisoblanadi.

Birinchi bob mavzuga kengaytirilgan kirishdan iborat. 1.1-bo'lim "ontologiya" tushunchasini o'rganish motivatsiyasini tavsiflaydi va ontologiyalardan foydalanmasdan ushbu vazifalarni hal qilish qiyin bo'lgan vazifalarning aniq misollarini beradi. 1.2-bo'lim ontologiyani belgilaydi va ushbu ta'rif bilan bog'liq bo'lgan turli tushunchalarni tavsiflaydi. 1.3-bo'lim har xil turdagi ontologiyalarning taqdimotiga bag'ishlangan. Ushbu bo'limda tasniflash va taksonomiyalarga misollar, shuningdek, munosabatlar ontologiyalari va rivojlanayotgan ontologiyalar kabi murakkabroq ontologiya turlari keltirilgan. Nihoyat, 1.4-bo'lim monografiyaning keyingi boblarida taqdim etiladigan narsalarning tavsifiga bag'ishlangan. Ikkinchi bob bir-biriga bog'liq bo'lmagan uchta bo'limdan iborat. 2.1-bo'limda kontseptualizatsiya kontseptsiyasi bilan bog'liq bo'lgan tushunchalarning falsafiy asoslari keltirilgan. Bu yerda prototip nazariyasi qisqacha ko'rinishda keltirilgan, shuningdek, kontseptual relyativizm deb ataladigan falsafiy yo'nalishning asoslari berilgan. Yuqorida aytilganlarga asoslanib, muhandislik muammosining kontseptualizatsiyasi natijasida kontseptual sxemaning ta'rifi berilgan. 2.2-bo'limda ontologiyani boshqarish dasturiy ta'minot tizimlarining ma'lum ilovalari haqida umumiy ma'lumot berilgan. Ushbu tizimlarning har biri umumiy tushunchalarning tavsifi sifatida o'zining yuqori darajadagi ontologiyasini o'z ichiga oladi. Bu ontologiyalar Guarino terminologiyasida kontseptual modellar emas, balki ontologiyalardir. Bo'limda, shuningdek, taniqli amerikalik tadqiqotchi Jon Sova tomonidan ishlab chiqilgan "OWL olmos" - yuqori darajadagi ontologiya tavsiflangan. Ontologiya tashqi tomondan qirrali olmosga o'xshash tushunchalar diagrammasi shaklida berilgan, shuning uchun nom. Va nihoyat, 2.3-bo'lim ontologiyalarni qurish uchun turli xil metodologiyalar to'plamidir. Uchta metodologiya ko'rib chiqiladi:

- UML tilini ishlab chiquvchilar tomonidan taklif qilingan metodologiya, dasturiy ta'minot tizimlarining modellarini qurish uchun maxsus yaratilgan.

- Gruber tomonidan "Ma'lumot almashish uchun ishlatiladigan ontologiyalar uchun taxminiy dizayn tamoyillari" maqolasida taklif qilingan metodologiya. Ushbu metodologiya ontologiyani qurishning bir qancha fundamental tamoyillariga asoslanadi. Ushbu kichik bo'lim ushbu tamoyillarning taqdimotiga bag'ishlangan.

· Ontologiyani rivojlantirishning BORO usuli deb ataladigan uslubga asoslangan metodologiya. Mavjud axborot tizimlarini axborot modellariga qayta tashkil etish muammosini hal qilish uchun BORO (Business Object Reference Ontology) metodologiyasi taklif qilingan. Ushbu bo'limda ushbu texnikaning tavsifi va uning asosiy tamoyillari keltirilgan.

1-bob OWL va tegishli tushunchalarga kirishga bag'ishlangan. Aslida OWL tiliga unchalik e'tibor berilmagan. Ushbu bobning asosiy maqsadi OWL tilida ontologiyalarni qurishni o'rgatish emas, balki o'quvchini ushbu tilni o'z ichiga olgan asosiy tushunchalar bilan tanishtirishdir. 1.1-bo'limda OWL amalga oshirish uchun mo'ljallangan Semantik Internetning asosiy tamoyillari keltirilgan. OWL XML sintaksisiga asoslangan, shuning uchun ushbu bo'limda ushbu tilning g'oyalari qisqacha tavsifi berilgan. 2.1-bo'limda OWLning yadrosi bo'lgan RDF asoslari ko'rsatilgan. Ushbu bo'lim, shuningdek, RDF omborlarini so'rash uchun SPARQL tilini tavsiflaydi. 2.2-bo'lim to'g'ridan-to'g'ri OWL tili bilan bog'liq bo'lgan tushunchalarning ekspozitsiyasiga bag'ishlangan. Ushbu bo'limda til asoslari, shuningdek, OWL tilining turli dialektlari haqida qisqacha ma'lumot berilgan. 2.3-bo'lim Protege muharririning interfeysi va takliflarini tavsiflaydi. Protege - OWL ontologiyasi muharriri. Bo'lim, shuningdek, bobda bayon etilgan g'oyalarni ko'rsatuvchi OWL da yozilgan ontologiyaning namunasi taqdim etadi. Bobning oxirgi qismida qiziqqan o'quvchi o'z tizimlarini ishlab chiqishda foydalanishi mumkin bo'lgan RDF omborlarining ba'zi "bepul" ochiq manba dasturlariga havolalar berilgan.

2-bobda Ontolingua ontologiyasini boshqarish tizimi bilan bog'liq turli g'oyalar bayon etilgan. 2.1-bo'lim tizimni yaratishga turtki bo'lgan g'oyalarni tavsiflashga bag'ishlangan. 2.2-bo'lim KIF (Knowledge Interchange Format) tilining tavsifini o'z ichiga oladi. Tavsif KIF tili uchun dpANS standarti loyihasiga asoslangan. Shunday qilib, ushbu bo'lim KIF til standartining norasmiy rus tilidagi versiyasi sifatida ishlatilishi mumkin. 2.3-bo'lim Ontolingua tizimida qo'llaniladigan KIF tilining versiyasining tavsifini o'z ichiga oladi. Ushbu dialekt asl KIFdan bir nechta farqlarga ega, bu farqlar ushbu bo'limda tavsiflangan. 4.4-bo'limda Ontolingua KIF tilida yozilgan ontologiyaning namunasi mavjud bo'lib, bobda keltirilgan g'oyalarni tavsiflash uchun berilgan. 2.5-bo'lim Ontolingua tizimi tomonidan

taqdim etilgan Ontologik ramka muharriri - ontologiyani tahrirlash vositasini tavsiflaydi.

Ramkalash

Tahrirlovchi veb-server sifatida o'z interfeysini shakllar bilan veb-sahifalar shaklida amalga oshiradi. Ushbu muharrir funksiyasining tavsifi ushbu bo'limning mazmunini tashkil qiladi. 2-bob Cyc bilimlarni boshqarish tizimida amalga oshirilgan g'oyalar taqdimotiga bag'ishlangan. Cyc tizimi birinchi darajali predikatlar hisobi tiliga asoslangan bilimlar bazasidir. 2.1-bo'lim tizimning asosiy xususiyatlarining tavsifiga va uning rivojlanish tarixiga bag'ishlangan. Bilimlar bazasini interfeys bilan ta'minlash, yangi bilimlarni qo'shish va ularni tartibga solish uchun Cyc da CycL deb nomlangan maxsus til amalga oshiriladi. Ushbu til 2.2-bo'limda tavsiflangan. 2.3-bo'limda Cyc bilimlar bazasida foydalaniladigan xulosalar tizimi tavsiflangan. Tizim xulosa chiqarishning asosiy usuli sifatida rezolyutsiyalar usulidan foydalanadi, ammo xulosa qilish algoritmini bajarish jarayonida variantlarni qidirishni kamaytirish uchun ko'plab evristik usullarni qo'llaydi. 2.4-bo'limda Cyc tizimining bilimlar bazasining yuqori darajadagi ontologiyasi tavsifi mavjud. Har qanday rivojlangan bilimlarni boshqarish tizimi singari, Cyc tizimi ularni qandaydir tarzda tizimlashtirishi kerak. Buning uchun Cyc ontologiyasi umumiy tushunchalarning murakkab tizimini, shuningdek, turli fan sohalari tushunchalari tizimini tavsiflaydi va ma'lum bir fan sohasi doirasida maxsus bilimlar qo'shiladi.

II BOB. AXBOROT TIZIMLARI ONTOLOGIYALARI

2.1. Ontologiyani qo'llashning turli jihatlar

Ontologiyaning metafizikasi

Ushbu bo'limda "kontseptual sxema", "ontologiya" va "kontseptualizatsiya" tushunchalarining asoslari bilan bog'liq turli masalalar muhokama qilinadi. Bu yerda biz loyqa falsafiy suvlarga sho'ng'ishimiz kerak. Afsuski, biz falsafasiz qilolmaymiz. Bu ehtiyoj o'rganilayotgan masalaning mohiyatidan kelib chiqadi: "kontseptsiya", "tur" yoki "sinf" atamaları, shuningdek, "kontseptsiya" atamaları falsafiy asoslashni talab qiladi, chunki ular bizni savolga bevosita murojaat qiladi. inson tafakkurining tashqi dunyo bilan qanday aloqasi borligi haqida. mohiyatan, falsafaning asosiy savoliga. Bir tomondan, bu ish falsafiy tadqiqot emas, shuning uchun aslida biz hikoya qilish jarayonida falsafiy muammolardan qochishga harakat qilamiz. Boshqa tomondan, hech bo'lmaganda bizning mavzimiz bilan bevosita bog'liq bo'lgan masalalarda o'z falsafiy kontseptsiyasiz qilolmaydi. Shunday qilib, ushbu bo'limning taqdimoti muallifga faylasuflarning adolatli g'azabining mahorati va o'quvchining noto'g'ri qarashlarining o'rtasidagi sayohatga o'xshash narsadek bo'lib tuyuladi.

Ammo, afsonaviy Odisseyadan farqli o'laroq, muallif hamon bu qiyin yo'lni yengishga umid qiladi. Klassik falsafada borliq ontologiyasi muammolarini muhokama qilganda, odatda, uning ob'ektiv tomonlari ko'rib chiqilar edi. Yangi falsafada bu muammoning sub'ektiv tomonlari ham ko'rib chiqiladi. Boshqacha qilib aytganda, nafaqat muhim narsa, balki unga qanday qarash ham muhimdir. Odatda, bu savollar muayyan ijtimoiy guruhlarni o'rganish jarayonida paydo bo'ladi. Bunday eng yirik ijtimoiy shakllanishlardan xalq (etnos) ni aytish mumkin. Taxminlarga ko'ra, har bir xalq u yoki bu darajada o'ziga xos bo'lish g'oyasiga ega, u qaysidir ma'noda boshqa xalqlarning g'oyalari to'g'ri keladi, garchi vulgar qarash umuman bunday sub'ektivizmni inkor etishdan iborat. Hozirgi vaziyat shundayki, ko'plab xalqlar Evropa sivilizatsiyasi madaniyatini o'zlashtirgan, shuning uchun turli xalqlar o'rtasida bo'lish haqidagi g'oyalardagi farqlar endi ahamiyatsiz. Biroq, bizning davrimizda u yoki bu sabablarga ko'ra Evropa madaniyati bilan tanish bo'lmagan xalqlar bor, keyin farqlar to'liq ko'rinadi.

Bu borada D.Devidson ajoyib misol keltiradi:

Whorf, Hopi tilining ingliz tili bilan "qiyoslab bo'lmaydigan" metafizikani o'z ichiga olganligini ko'rsatmoqchi bo'lib, Hopi jumllalarining mazmunini etkazish uchun ikkinchisidan foydalanadi!

Borliq haqida maxsus g'oyalar mavjudligi haqidagi g'oyani konseptual relativizm deb atash mumkin. Biz kontseptual relyativizm doirasini kengaytiramiz, mavjudlikni ko'rib chiqishning sub'ektivligini har qanday muhandislik muammosini hal qilish uchun shakllanadigan kichikroq ijtimoiy guruhlar bilan bog'laymiz. Ushbu muammoni hal qiladigan guruh ushbu vazifaning maqsadi bilan shartlangan o'ziga xos nuqtai nazardan qaraydi deb taxmin qilinadi. Bu nuqtai nazar ko'rib chiqish doirasidan faqat ma'lum bir vazifa uchun zarur bo'lgan tafsilotlar va ob'ektlarni ajratib turadi va qolgan hamma narsani e'tiborsiz qoldiradi. Olingan maxsus voqelik g'oyasini ma'lum bir vazifa bilan bog'liq bo'lgan ijtimoiy guruhning ontologiyasi yoki ma'lum bir guruhdan mavhum holda, vazifaning ontologiyasi deb atash mumkin. Ushbu ish muhandislik muammolari ontologiyalarini, xususan, ularni rasmiy tavsiflashning turli usullarini o'rganishga bag'ishlangan.

Ammo, ushbu usullarni ko'rib chiqishga o'tishdan oldin, har bir bunday o'ziga xos borliq g'oyasi nima ekanligini, u nimadan iboratligini va qanday tuzilganligini aniqlash kerak. Bu savollarga javob berish uchun falsafaga murojaat qilaylik, xususan, faylasuflar borliq ontologiyasining tuzilishini uning ob'ektiv jihatlarida ko'rib chiqilishini qanday ifodalaydilar. Ehtimol, borliqning tuzilishi haqidagi tushunchalarni shakllantirishning birinchi manbasini til deb atash mumkin. Tilda odamlar bir-biri bilan muloqot qiladilar va xususan, o'zlarini qiziqtirgan borliq sohalarining tavsiflarini shakllantiradilar. Til so'zlardan iborat, ya'ni diskret. Shuning uchun so'zlarni odamlar o'zlarining borliq g'oyalarini tavsiflaydigan asosiy elementlar deb atash mumkin. E'tibor bering, bir xil so'z turli xil ma'nolarni anglatishi mumkin. Masalan, "stul" so'zi o'zining funktsional maqsadi jihatidan bir xilda ko'rib chiqiladigan juda ko'p turli xil ma'nolarni anglatishi mumkin. Demak, "stul" so'zi aslida bir narsani emas, balki butun bir tabaqani (majmuini) bildiradi. Bunda biz "sinf" atamasidan foydalanamiz. Agar siz berilgan narsaning qaysi sinfga bog'liqligi haqida savol bersangiz, biz "tur" atamasiga kelamiz. Tur narsaning o'ziga xos belgilari deb ataladi, ular asosida biz berilgan

narsani berilgan sinf bilan bog'laymiz. Bu o'ziga xos xususiyatlar ma'lum bir nuqtai nazardan qaralganda, ma'lum bir sinfga tegishli barcha narsalar uchun bir xil bo'ladi, deb taxmin qilinadi. Bu yerda shuni yodda tutish kerakki, narsalarning turlari mavjud bo'lib, ular asosida ular sinflarga shakllanadi, faqat qandaydir maqsad tufayli qandaydir nuqtai nazar mavjud bo'ladi, ular nuqtai nazaridan narsalar ma'lum bir o'ziga xoslikka ega bo'ladi. Shunday qilib, agar biz o'tirish uchun narsalarni tanlash uchun emas, balki ularni yoqish uchun bo'lishni nazarda tutsak, temir va yog'och stullar turli sinflarga tegishli bo'lar edi. Biroq, turli maqsadlar o'zaro bog'liq bo'ladi - bir maqsad boshqasining bir qismidir. Masalan, men o'z kvartiramda yashash uchun kerakli narsalar to'plamini olmoqchiman. O'tirish uchun menga stol kerak, ovqatlanish uchun men biron joyga ovqat qo'yishim kerak, ya'ni menga stol kerak, uxlash uchun to'shak kerak. Bu yerda hamma narsa funktsional nuqtai nazardan ko'rib chiqiladi, lekin har bir narsaning funktsiyalari biroz o'xshash bo'lsa-da, farq qiladi. Bularning barchasini bitta atama deb atash uchun men "mebel" so'zini ishlataman. Shunday qilib, yangi narsalar sinfi paydo bo'ladi, u "stul" atamasi bilan belgilanadigan sinfni o'z ichiga oladi.

Shunday qilib, sinflar va shuning uchun bu sinflarga mos keladigan turlar ierarxiyani tashkil qilishi mumkin, bunda aniqroq sinfning har bir narsasi ham umumiy sinfning narsasidir. Buni ham aytishimiz mumkin: ko'rib chiqish maqsadi sinflar va ularning tegishli turlari ierarxiyasini tashkil qiladi. Eng taxminiy shaklda bunday ierarxiyani borliqni ifodalash shakli deb hisoblash mumkin va bu tavsiflash ushbu mulohaza amalga oshiriladigan nuqtai nazardan vazifa maqsadi asosida shakllanadi. Lekin, albatta, masala faqat narsalar turlarining ierarxiyasi bilan chegaralanib qolmaydi, turlar orasidagi ierarxiyadan tashqari, boshqa turdagi munosabatlar ham mavjud bo'lishi mumkin va ko'pincha bu munosabatlar juda murakkab va chalkashdir. Vazifaning maqsadi bilan belgilanadigan nuqtai nazar ham ko'rib chiqish konteksti deb ataladi. Oldingi paragrafda tavsiflangan vazifa ontologiyasini tuzishga yondashuv bilan bog'liq terminologiyani ko'rib chiqing. Quyida keltirilgan atamalar asosida biz keyingi rivoyatni tuzamiz.

Sinf

Yuqorida aytib o'tilganidek, sinf bu ko'rib chiqilayotgan muammoning maqsadi nuqtai nazaridan bir xil tarzda ko'rib

chiqiladigan narsalar yig'indisidir. Shu bilan birga, berilgan ob'ektni ma'lum sinfga mansublik vazifasi ba'zi hollarda aniq bo'lishi mumkin, u juda qiyin va chuqur izlanishni talab qiladi, ba'zan esa noaniq bo'lib, bir ma'noli yechimga yo'l qo'ymaydi. lekin shartnomalar predmeti hisoblanadi.

Tur

Tur narsalarning farqlovchi belgilarini ifodalaydi. Ob'ektlar ko'rib chiqilayotgan vazifaning maqsadiga qarab berilgan sinfga guruhlangan. Bu o'ziga xos xususiyatlar ma'lum bir sinfga tegishli barcha narsalar uchun bir xil bo'ladi, aks holda narsalar bir sinf bilan bog'lanmaydi. Narsalarning o'z-o'zidan turlari bo'lmaydi, lekin narsalarning turlari faqat berilgan vazifaning maqsadi bilan belgilanadigan ma'lum nuqtai nazardan qaralganda paydo bo'ladi. Turlar odatda ierarxiyaga ajratiladi, unda aniqroq turdagi narsalar ham umumiy turdagi narsalardir. Terminologiya typeSubtype ham qo'llaniladi.

Kontseptsiya

Tushuncha, agar u ko'rib chiqish nuqtai nazaridan va shu nuqtai nazar bilan belgilanadigan narsalarning xususiyatlarini hisobga olmagan holda ko'rib chiqilsa, bu tipdir. Bunday yondashuv, masalan, tadqiqotchi odatda muammolarning maqsadlaridan abstraktatsiya qiladigan va nima uchun kiritilganligidan qat'i nazar, tushunchalar orasidagi turli mumkin bo'lgan munosabatlarni ko'rib chiqadigan matematik modellarda mantiqiydir. Keling, yuqorida keltirilgan ta'riflarni oddiy misol bilan tushuntirib beraylik. Keling, "quyon" tushunchasini ko'rib chiqaylik. Biz hozir qo'llagan "kontseptsiya" atamasining o'zi "quyon" endi mavhum tarzda ko'rib chiqilishini anglatadi, ya'ni har qanday kontekstdan qat'i nazar. Shunday qilib, biz "quyon" deymiz - bu har qanday kontekstdan tashqarida, umuman olganda, quyon haqida gapiradigan bo'lsak, bu "tushuncha" va shuning uchun biz bu tushunchani biron bir tarzda aniqlashtirishimiz shart emas. Tur, ta'bir joiz bo'lsa, ma'lum bir kontekstga singib ketgan va shuning uchun uni ushbu kontekstning boshqa tushunchalaridan ajratib turadigan xususiyatlarga va ushbu kontekstning boshqa turlari bilan aloqalariga ega bo'lgan nozik tushunchadir. Masalan, "ov quyoni" turi "hayvonot bog'i quyoni" turidan turli xossalari va boshqa turlar bilan munosabatlari bilan farq qiladi. "Ovdagi quyon" turi uchun uning yugurish tezligi, izlarni xiralashtirish qobiliyati, shuningdek,

"odam", "it", "tulki" va boshqa ovchilar turlari bilan bog'lanish kabi xususiyatlar muhim ahamiyatga ega. "Hayvonot bog'idagi quyon" turi uchun yuqoridagi xususiyatlar muhim emas, lekin ular muhim, masalan, "xarakterning mosligi", "go'zallik", "infektsiyaga qarshilik" va boshqalar. Sinf - bu ma'lum turdagi misollar to'plami. Bizning holatda, bu hayvonot bog'idagi quyonlar yoki o'rmondagi quyonlar bo'lishi mumkin. Sinf to'plam emas, chunki sinfnig barcha holatlari har doim ham ma'lum emas (agar hayvonot bog'ida qancha quyon borligini hali ham aniqlash mumkin bo'lsa, o'rmonda qancha quyon borligini tushunish allaqachon muammoli). Arktik doiradagi tuyaqushlar sinfini darslik misolida keltirish mumkin. Biz katta ehtimollik bilan bu sinfnig bo'sh ekanligini aytishimiz mumkin, ammo bunga to'liq ishonch hosil qilish mumkin emas, chunki Shimoliy Muz okeanining biron bir joyida suzib ketayotgan kemada tuyaqush yo'qligiga hech kim kafolat bera olmaydi.

Turkum

Turlarning o'ziga xos xususiyati shundaki, muammoning maqsadlariga qarab, ularning o'zlari sinflarga bo'linishi mumkin. Keyin turlarga yangi turlar tayinlanadi, ular asosida ular sinflarga birlashtiriladi, ya'ni bu mulohaza kontekstidagi turlar narsa sifatida namoyon bo'la boshlaydi. Yangi matematik konstruksiyalarning kiritilishi turlarning shu xossasiga asoslanadi. Ko'rib chiqishning yangi kontekstida "geometriya" tushunchasi narsaga aylanadi - Evklid geometriyasi, ko'plab mumkin bo'lgan geometriyalardan biri. Butun sonlar, degenerativ bo'lmagan matritsalar va elementlarning almashinuvi kabi narsalarning farqlovchi xususiyatlarini ifodalovchi tip guruhi nazariy jihatdan narsaga aylanadi.

Algebrada ontologiya

Falsafada turlar ierarxiyasi ham borliq ontologiyasi bilan bog'liq. Ushbu ierarxiyaning eng yuqori qismida eng keng tarqalgan turlar joylashgan. Agar tur boshqasining kichik turi bo'lsa, uni shunday aniqlash mumkin: falon tip qo'shimcha funktsiyalarga ega tur. Bu umumiy tushuncha yordamida bitta tushunchani aniqlashning odatiy usuli. Eng umumiy turlar uchun bunday ta'rif mumkin emas. Bunday tushunchalar odatda kategoriyalar deb ataladi. Qaysi aniq tushunchalar kategoriyalar deb hisoblanishi aniq falsafiy tushunchaga bog'liq. Birinchi marta "kategoriya" atamasi Aristotelda paydo bo'lib, u o'nta kategoriyani aniqlagan.

Keyinchalik bu atama Kant tomonidan o'zining "transsendental mantiq"ida aprior bilim deb ataladigan bilimni ifodalash uchun ishlatilgan. Kant fikricha, hislar va idroklar olami tartibsiz, tartibsiz tuyg'ular va hodisalar to'plami bo'lib, uning makon va vaqt kabi sezuvchanlikning aprior shakllari tartiblangan. Fazo va vaqt real dunyoda mavjud emas, faqat tashqi olamni idrok etish shakllarini ifodalaydi. Fazo va vaqt haqiqatda mavjud bo'lmagani uchun ular haqidagi bilimlarni har qanday tajribadan oldingi va shuning uchun apriori bo'lgan tajribadan olish mumkin emas. Fazo va vaqt prizmasi orqali idrok etiladigan hodisalarga bog'lanishlarni o'rnatish aql deb ataladigan kategoriyalar orqali sodir bo'ladi. Bularga miqdor kategoriyalari, ya'ni birlik, ko'plik va yaxlitlik, sifat kategoriyalari: voqelik, inkor va cheklov kabilar kiradi. Kantdagi tushuncha kategoriyalarining kamchiligi shundaki, ular tushuncha faoliyatidan kelib chiqmaydi, balki Kantning o'zi tomonidan o'zboshimchalik bilan belgilab qo'yilgan. Bu kamchilik Gegel falsafasida dialektika yordamida hal qilinadi, u orqali Gegel borliqning o'zidan aql kategoriyalarini «chiqarib oladi».

Ma'no ontologiyalari

Ma'no ostida biz psixologik hodisani, o'ziga xos ruhiy holatni tushunamiz, uni "tushunish" atamasi bilan ham ifodalash mumkin. Ammo tushunish bu holatni umumiy tarzda tavsiflaydi va ma'no muayyan narsa bilan bog'liq. Boshqacha qilib aytganda, biror narsaning ma'nosi - bu alohida shaxsning o'ziga xos psixologik holatida ifodalangan, "Men tushunaman" iborasi bilan ifodalanishi mumkin bo'lgan ma'lum bir ruhiy holat.

Ma'nolar

Ma'nolar - bu turli shaxslarning ma'lum bir narsaga taxminan teng munosabatda bo'lgan ma'nolari (qoida tariqasida, bu kelishuv natijasida erishiladi). Ma'nolar - ma'nolarning abstraktsiyalari - individual tajribalar kontekstidan ajratilgan tushunchalar.

Ob'ekt

Ko'rib chiqish kontekstiga qarab, ob'ekt ham turni, ham qandaydir narsani ifodalashi mumkin. Odatda, "ob'ekt" atamasi ma'lum bir shaxsning ko'rib chiqish doirasini ajratib ko'rsatish uchun ishlatiladi, bu narsa narsa yoki tur ekanligini aniqlamaydi.

Misol

Misollar sinf a'zolaridir. Odatda, misollar narsalardir, lekin yuqorida muhokama qilinganidek, boshqa kontekstda narsalar turlarini narsalar deb hisoblash mumkin. Narsalarning sinflarga birlashtirilganligi, uning turi bilan bog'lanish mexanizmi qanday, tipning umumiy tuzilishi nimadan iboratligini aniq belgilab qo'yuvchi xususiyatlar nima ekanligini hali aniqlab olganimiz yo'q. Bu savolga javobni Aristotel ham taklif qilgan. Aristotelning fikricha, har bir tur uning o'ziga xos xususiyatlari (belgilari yoki atributlari) to'plamiga mos keladi.⁴ Atributlar - bu ko'rib chiqishning berilgan kontekstida muhim bo'lgan narsaning turli xil xususiyatlari. Atributlar muhim va muhim bo'lmagan deb tasniflanadi. Muhim atributlar - bu narsaning qaysi sinfga mansubligini aniqlash imkonini beruvchi xossalari, ya'ni ushbu turdagi o'ziga xos xususiyatlar. Muhim bo'lmagan atributlar - bu ma'lum turdagi barcha narsalarga ega bo'lgan, lekin biror narsani berilgan sinfga belgilashda qatnashmaydigan xususiyatlar.

Masalan, "olma" tushunchasi uchun muhim xususiyatlar bo'ladi: meva (kelib chiqishi), yumaloq (shakli), ta'mi (barcha olma uchun umumiy narsa) va boshqalar. Arzimas xususiyatlar rang (olma turli xil ranglarda bo'ladi), hajmi (rezavorlarga o'xshash samoviy olma bor), vazn va boshqalar bo'ladi. Hozirda ommabop dasturlashning ob'ektga yo'naltirilgan yondashuvi ham Aristotel an'analariga asoslanadi. Masalan, C++ dasturlash tilida turlar class kalit so'zi yordamida ko'rsatiladi, undan keyin sinf atributlarining turlari va nomlari, shuningdek uning usullari sanab o'tiladi. Ushbu sinfning o'ziga xos namunasi atribut xususiyatlarining o'ziga xos qiymatlarini o'z ichiga oladi va aslida uning xususiyatlari to'plamidan boshqa narsa emas. Shuni ta'kidlash kerakki, asosiy turdagi xususiyatlar C ++ sinfini qurishda ko'rsatilmagan, chunki ular har doim sinfning barcha misollari uchun bir xil qiymatlarga ega. Ushbu qiymatlar barcha sinflarda asosiy sinfning tegishli atributlarining sobit qiymatlari sifatida mavjud. Aristotelning narsalarning turlarini ko'rsatish an'anasi bizning zamonamizda asosiy hisoblanadi, ammo turlarni belgilashda tubdan farqli yondashuvlar ham mavjud. O'tgan asrning 70-yillarida kognitiv semantikada boshqacha yondashuv taklif qilindi. Uning muallifi Kaliforniyaning Berkli shahridan psixologiya professori **Eleanor Rosh** hisoblanadi. Kontseptsiyalarni farqlashning bunday yondashuvini tavsiflovchi nazariya prototip nazariyasi deb ataladi.

Ushbu nazariyaning asosiy g'oyasi har bir turdagi ushbu kontseptsiyaning prototipi deb ataladigan kamida bitta xarakterli vakiliga ega degan taxminga asoslanadi. Keyin, berilgan narsa ma'lum bir turga tegishlimi yoki yo'qmi, bu narsaning ushbu turdagi prototipga o'xshashlik darajasini tekshirish mumkin. Albatta, har bir kontseptsiya uchun hali ham uning prototipini ajratib ko'rsatish kerak va katta muammo bu narsa ma'lum bir turdagi tipik vakili yoki yo'qligini aniqlashdir.

Rosh ma'lum bir narsa kontseptsiya prototipi ekanligini ko'rsatadigan quyidagi ko'rsatkichlarni taklif qiladi:

1. Javob berish vaqti. Berilgan kontseptsiya prototipi uchun ma'lum turdagi narsalarni nomlash haqidagi savolga javob berish vaqti atipik elementlarga qaraganda tezroq.

2. Oldindan ulanish. Agar siz avval mavzuni yuqori darajadagi biron bir tushuncha bilan bog'lasangiz, ikkita so'zning bir xil ekanligini tan olish tezroq bo'ladi. Misol uchun, agar siz avval "asbob" yozuvi bo'lgan belgini ko'rsatsangiz, quyida ko'rsatilgan ikkita planshetda bir xil "bolg'a" so'zi yozilganligini tan olish "maydalagich" so'zini tanib olishdan ko'ra tezroq sodir bo'ladi.

3. Misollar. Agar suhbatdoshdan kontseptsiyaning bir nechta misollarini nomlashni so'rasangiz, prototiplar tez-tez nomlanadi. Albatta, o'quvchi shunday o'yin o'ynagan. Taxmin qilgan kishi avval qog'ozga uchta so'z yozadi, so'ngra sherikdan ortga hisoblashni yuzdan boshlashni so'raydi. Sherik hisob bilan chalg'igan vaqtni kutgandan so'ng, iltimos qiluvchi so'raydi:

1. Har qanday shoirning nomini ayting.

2. Har qanday mevani nomlang.

3. Har qanday asbobni nomlang.

Sherik buni qilgandan so'ng, so'rovchi ilgari qog'ozga yozgan javoblarini g'alaba bilan ko'rsatadi: "Pushkin", "olma" va "bolg'a". Shubhasiz, aslida, shu tarzda tegishli tushunchalarning prototiplari ajralib turadi. Aristotelcha yondashuvdan farqli o'laroq, sinfning barcha holatlari teng ravishda ifodalanadi, prototip nazariyasida misolning sinfga muvofiqligi o'lchovi mavjud. Shubhasiz, sinfga eng yaqin misollar prototiplardir. Ushbu misollar kontseptsiyaning "markazida" joylashgan. Boshqa namunalar "bir oz uzoqroqda" joylashgan. Kontseptsiya bir nechta deyarli ekvivalent prototiplarga ega bo'lishi mumkin. Masalan, "Lermontov" misoli "shoir"

tushunchasining prototipi vazifasini ham bajarishi mumkin, “o‘rik” esa “meva” tushunchasini ifodalash uchun ham ishlatilishi mumkin. Misolning ma'lum bir sinfga tegishlilik darajasini ma'lum bir misoldan kontseptsiya sinfining prototipiga o'xshashligi bilan baholash mumkin. Ma'lum bo'lishicha, biz biroz keyinroq ko'rib chiqamiz, misolning ushbu misol sinfiga mosligi o'lchovi metrik, ya'ni metrik funktsiyaning matematik ta'rifi uchun zarur bo'lgan barcha rasmiy xususiyatlarni qanoatlantiradi.

Prototip nazariyasi bilan bog'liq bo'lgan va Roche tomonidan taklif qilingan yana bir yangi yondashuv - bu asosiy tushunchalar deb ataladigan usul. Bunday tushunchalar turlar ierarxiyasida aniqlanadi va odatda qandaydir vosita yoki hissiy harakatlar bilan bog'liq. Misol uchun, "Siz nima o'tiribsiz?" odatda javob "ofis kreslosida", "oshxona stulida" yoki oddiygina "mebelda" emas, balki "stulda". Kreslo, tushuncha sifatida, uning asosiy motor funktsiyasi bilan bog'liq - tizzalarning egilishi. Bundan tashqari, stul chizilgan bo'lishi mumkin bo'lgan eng yuqori darajadagi tushunchadir. "Mebel" tushunchasining keyingi darajasi uchun bu endi mumkin emas. Aristotelchilarning tipni atribut-qiyamat juftliklari yig'indisi sifatida tushunishga asoslangan an'ana uchun bu asosda asosiy daraja tushunchasini ajratib ko'rsatish qiyin. Funktsional jihatdan aytishimiz mumkinki, asosiy darajadagi tushunchalar voqelikning eng informatsion toifalarga bo'linishini ifodalaydi.

- Berilgan kontseptsiyaning barcha vakillari ega bo'lgan atributlar sonini maksimal darajada oshirish.

- Boshqa tushunchalarning barcha vakillari ega bo'lgan atributlar sonini minimallashtirish.

Prototip nazariyasi rasmiy ravishda quyidagicha taqdim etilishi mumkin. Har bir kontseptsiya c uchun uning prototipi p ajratiladi va maxsus funksiya $d(x, y, c)$ o'rnatiladi - ob'yektning x ob'yektdan y semantik masofasi, agar ularni c tipi vakillari deb hisoblasak. Keyin x ob'ektini quyidagi rekursiv protsedura bo'yicha tasniflash mumkin:

1. Faraz qilaylik, x ob'ekti s_1, s_2 , kichik turlariga ega bo'lgan c tipidagi misol sifatida allaqachon tan olingan., s_n .

2. Har bir kichik tip s_i uchun biz semantik masofani $d(x, p_i, c)$ o'lchaymiz, bunda p_i s_i kenja turi vakili.

3. Agar shunday barcha semantik masofalar $d(x, p_i, c)$ orasida barcha boshqalardan qat'iy kam bo'lgan qiymat bo'lsa (noyob minimum), u holda $c = s_i$ ni o'rnatamiz va boshidan boshlaymiz.

4. Agar shunday barcha semantik masofalar $d(x, p_i, c)$ barcha boshqalaridan qat'iy kam bo'lgan qiymatga ega bo'lmasa, u holda tasniflash jarayoni to'xtaydi va x ob'yekt c tipidagi hisoblanadi.

Misol uchun, ikkita ob'ektni ko'rib chiqing: qora mushuk va zanjabil mushuk. "Hayvonlar" sinfida bu ikki ob'ekt juda o'xshash va deyarli farq qilmaydi, ammo "Mushuklar" sinfida bu ob'ektlar orasidagi farq allaqachon sezilarliroqdir, chunki ob'ektlar mushuklarga xos bo'lgan barcha xususiyatlarga ega. Boshqa tomondan, agar "Qora buyum" turini ko'rib chiqsak, undagi "qora mushuk" obyektiga "zanjabil mushuk" obyektiga qaraganda ko'proq "qarg'a" yoki "ko'mir bo'lagi" obyektiga o'xshaydi.

Tushunchalarni farqlashning tavsiflangan usuli aniq misollarga asoslanganligini inobatga olib, ma'no masofasining vazifasini ham ana shunday misollar asosida aniqlash mumkin. Shu sababli, ushbu turdagi ta'rif, ayniqsa, klaster tahlili va neyron tarmoqlar kabi statistikaga asoslangan turli xil ilovalarda mashhur. 2000 yilda Lund universiteti (Shvetsiya) professori Piter Gardenfors tomonidan "Konseptual fazolar" monografiyasi nashr etildi. Ushbu monografiyada Gardenfors tushunchalarni ifodalash uchun geometrik yondashuvni taklif qildi. Ushbu yondashuvda tushunchalarning xossalari kontseptual fazoning o'lchamlari ko'rinishida ifodalanadi, bu xususiyatlarning dekart mahsuloti bilan olingan va ma'lum bir metrik bilan jihozlangan. Har bir kontseptsiya kosmosning konveks mintaqasini ifodalaydi. Kontseptsiyaning o'ziga xos misollari ma'lum bir sohada nuqta sifatida ifodalanadi. Oldingi xatboshida biz "semantik" deb atagan misollar orasidagi masofa aniqlanadi. Ba'zi xususiyatlar bir-biri bilan o'zaro bog'langan, ya'ni inson ongida bir-biridan ajralmas. Masalan, xroma, yorqinlik va to'yinganlik kabi rang xususiyatlari ajralmasdir. Gardenforce bunday xususiyatlarni mavzular deb ataladigan o'z joylariga birlashtiradi. Xususiyatlarni mavzularga birlashtirish inson tafakkurini to'g'ri modellashtirish uchun juda muhimdir. inson ongida bir-biridan ajralmas xususiyatlarni ajratish noto'g'ri modellarga olib keladi.

Gardenfors insonning fikrlash jarayonini ifodalashning uchta darajasini ajratib ko'rsatadi:

1. Fikrlash qobiliyatini ta'minlovchi biologik mexanizmlar ko'rib chiqiladigan asosiy daraja. Ushbu mexanizmlar odatda "neyron tarmoqlar" deb ataladigan modellar orqali modellashtiriladi.

2. Geometrik daraja, bunda fikrlash allaqachon tushunchalar yordamida ifodalanadi, ya'ni yuqorida tavsiflanganidek. Ammo kontseptual fazolarning geometriyasi shundayki, tegishli xususiyatlar vakillikning bu darajasida bir-biridan ajralmasdir.

3. Simvolik daraja, bunda tushunchalar va ularning xossalari ramziy ifodalar yordamida tavsiflanadi, xossalar va tushunchalar o'rtasidagi bog'lanishlar esa chegaralovchi gaplar (aksiomalar) yordamida o'rnatiladi. Taqdimotning uchta darajasi ham bir-biriga bog'langan va har biri o'z maqsadlari uchun ishlatiladi.

Gardenfors bu aloqalarni tushuntirish uchun "jungle" metaforasini ishlatadi. Aytaylik, sayohatchi keyingi yo'lni bilmasa, o'rmonda o'zini topdi. U aborigendan so'radi va unga keyingi yo'lni quyidagicha tushuntiradi: "To'g'ri uch yuz qadam yur, keyin o'ngga burilib, yana ikki yuz qadam yur, so'ng chapga burilib, katta yo'lga chiqquncha bor." Bunday vaziyatda ushbu tushuntirish avtomagistralga marshrut haqidagi ma'lumotni etkazishning eng yaxshi usuli hisoblanadi va marshrutning bu tavsifi asosiy hisoblanadi. Aytaylik, magistralga boradigan yo'l juda uzun va ko'p burilishlarni talab qiladi, qadamlar sonini yodlaydi va hokazo. narsalardan. Bunday holda, bunday tushuntirish ishlamaydi, chunki shunchaki sayohatchini yanada chalg'itadi. Sayohatchiga mazmunli nishonlarni berish kerak. Boshqacha qilib aytganda, bu holatda avtomagistralga boradigan yo'lning yaxshi ta'rifi shunday ko'rinishga ega bo'ladi: "to'g'ridan-to'g'ri bo'yimdan pastroqda bo'shliqli baobabga duch kelmaguningizcha, keyin o'ngga buriling va daryo qirg'og'iga boring va Tuproq yo'lga duch kelmaguningizcha uni chapga kuzatib boring, shunda siz u bo'ylab o'ngga burilib, katta yo'lga yetasiz." Yo'lning bu tavsifi geometrik darajaga to'g'ri keladi, allaqachon tushunchalar (baobab, daryo, yo'l) mavjud, ammo ular bir-biriga bog'langan, o'rmonga tabiiy tarzda bog'langan (ular ichida bo'lish). Agar sayohatchining xaritasi bo'lsa, unda aborigendan unga kerak bo'lgan yagona narsa bu sayohatchining hozirda joylashgan joyini va asosiy nuqtalarning yo'nalishini ko'rsatishdir. Bu sayohatchining katta yo'lni

o'zi topishi uchun yetarli bo'ladi. Xarita ramziy vakillik darajasiga mos keladi. Xaritada er relyefi xarita masshtabiga mos ravishda aniq belgilanadi, hudud uchun diqqatga sazovor joylar ajratiladi, bu hududlarga nomlar beriladi va hokazo.

Albatta, qonuniy savol tug'ilishi mumkin: agar Aristotelcha nazariya mavjud bo'lsa, nima uchun kontseptsiyani ifodalashning yangi nazariyalaridan foydalanish kerak? Aristotel an'alariga ko'ra atributlardan foydalanadigan turlarning ta'rifi juda yaxshi ishlaydi, shuning uchun kontseptsiyaning yangi nazariyasiga akademikdan boshqa qiziqish mavjudligi aniq emasmi? Birinchidan, yangi nazariyalar inson tafakkurining Aristotel nazariyasi qamrab ololmaydigan ayrim tomonlarini aniqroq aniqlash imkonini beradi. Shu ma'noda, Aristotelcha tushunchalarni ifodalash nazariyasi xuddi Nyutonning tortishish nazariyasi umumiy nisbiylik nazariyasiga taalluqli bo'lganidek, tushunchalarning yangi nazariyalariga ham ishora qiladi. Bundan tashqari, har bir kontseptsiyani uning xususiyatlarini belgilash orqali aniqlash mumkin emas. Keling, Vitgenshteynning mashhur "Falsafiy izlanishlari" dan bir misol keltiraylik.

Masalan, biz "o'yinlar" deb ataydigan jarayonlarni ko'rib chiqaylik. Men stol o'yinlari, karta o'yinlari, to'p o'yinlari, kurash va hokazolarni nazarda tutyapman. Ularning barchasida qanday umumiylik bor? "Ular uchun umumiy narsa bo'lishi kerak, aks holda ularni o'yin deb atashmaydi" demang, balki ularning barchasi uchun umumiy narsa bor-yo'qligini ko'ring. Darhaqiqat, ularga qarab, siz ularning barchasiga xos bo'lgan umumiy narsani ko'rmaysiz, lekin siz o'xshashlik, qarindoshlik va bundan tashqari, bir qator umumiy xususiyatlarni ko'rasiz. Yuqorida aytib o'tilganidek: o'ylamang, lekin qarang! Misol uchun, bir nechta yaqinliklarga ega stol o'yinlariga qarang. Keyin karta o'yinlariga o'ting: bu yerda siz birinchi guruh o'yinlari bilan ko'plab o'xshashliklarni topasiz. Ammo ko'p o'xshashliklar yo'qoladi va boshqalar paydo bo'ladi. Agar biz hozir to'p o'yinlariga murojaat qilsak, unda juda ko'p o'xshashliklar qoladi, lekin ko'p narsa yo'qoladi. Ularning barchasi "ko'ngilochar"mi? Shaxmatni yo'q va xoch o'yini bilan solishtiring. Barcha o'yinlarda g'alaba va mag'lubiyat bormi, har doim o'yinchilar o'rtasida raqobat elementi bormi? Jungle o'yinlari haqida o'ylab ko'ring. To'p o'yinlarida g'alaba va mag'lubiyat bor. Ammo bolaning devorga to'p tashlab, uni

ushlab olish o'yinida bu belgi yo'q. San'at va omad qanday rol o'ynashini ko'ring. Shaxmat va tennisda mahorat qanchalik farq qiladi. Dumaloq raqslar haqida o'ylab ko'ring! Bu yerda, albatta, o'yin-kulgi elementi bor, lekin qancha boshqa xususiyatlar yo'qolib bormoqda. Shunday qilib, biz ko'p, ko'p turdagi o'yinlarni ko'rib chiqishimiz mumkin, ular orasidagi o'xshashliklarning kelishi va ketishini kuzatamiz.

Shunday qilib, "o'yin" tushunchasi uchun biz "o'yin" nomini beradigan barcha ob'ektlar tomonidan umumiy bo'lgan xususiyatlar to'plamini ajratib ko'rsatish juda qiyin, agar umuman mumkin bo'lmasa. Boshqa tomondan, ushbu kontseptsiyaning bir nechta prototiplarini oddiy sanab o'tish sizga "o'yin" tushunchasining ma'nosini boshqa odamga aniq etkazish imkonini beradi. **Vitgenshteyn** bu haqda nima deydi:

Xo'sh, kimgadir o'yin nima ekanligini qanday tushuntira olasiz? O'ylaymanki, men unga o'yinlarni tavsiflab berishim kerak va bunga qo'shimcha qilishim kerak: "Bu va shunga o'xshash narsalar unga o'yinlar deyiladi". Biz o'zimiz bundan ko'proq narsani bilamizmi? O'yin nima ekanligini boshqalarga ayta olmaymizmi? Lekin bu bilimsizlik emas. Biz o'yin tushunchasining chegaralarini bilmaymiz, chunki ular o'rnatilmagan. Yuqorida aytib o'tilganidek, biz ba'zi maxsus maqsadlar uchun ma'lum bir chegara chizishimiz mumkin. Bu faqat endigina ushbu kontseptsiyadan foydalanish mumkinligini anglatadimi? Umuman yo `q! Agar ma'lum bir maxsus maqsad bo'lmasa. Xuddi shu darajada, "1 qadam = 75 sm" ta'rifi "1 qadam" uzunlik o'lchovini kiritadi. Agar siz menga e'tiroz bildirmoqchi bo'lsangiz: "Ammo bundan oldin bu uzunlikning aniq o'lchovi emas edi", men javob beraman: demak, bu noto'g'ri edi. Garchi siz hali ham menga aniqlik ta'rifi uchun qarzdor bo'lsangiz ham.

Kontseptual relativizm odatda ma'lum bir ijtimoiy guruh a'zolari tomonidan bo'lishning alohida nuqtai nazarini bildirish uchun "ontologiya" atamasidan foydalanmaydi. Buning uchun "kontseptual sxema" atamasi qo'llaniladi. Kontseptual relativizmدا kontseptual sxema hissiy ma'lumotlarga shakl beradigan aniq toifalar tizimi asosida tajribani tashkil qilish usuli sifatida tushuniladi. Kontseptual sxema bo'lib o'tayotgan voqealarga shaxslar, madaniyatlar va davrlarning nuqtai nazarini ifodalaydi. Bizning kontseptual relativizmni talqin qilishda, har bir kontseptual sxemaning orqasida

borliq nuqtai nazarini belgilovchi, shuningdek, ko'rib chiqish konteksti deb ataladigan maqsad mavjud. Ko'rib chiqishning muayyan kontekstida faqat berilgan vazifaning maqsadi bilan bog'liq bo'lgan narsalarning ba'zi jihatlari hisobga olinadi, ya'ni narsalar teriladi va shu yozish asosida sinflarga birlashtiriladi. Narsalarning turlari o'rtasidagi munosabatlar belgilanadi. Keyin, kontseptual sxema ana shunday turlar va ular o'rtasidagi munosabatlarning yig'indisi bo'ladi. Shu bilan birga, tiplashtirish Aristotel an'analari asosida yoki boshqa yo'l bilan qanday amalga oshirilganligi aniqlanmagan. Asosiysi, vazifaning maxsus kontseptual sxemasining mavjudligi, bu vazifaning maqsadi bilan belgilanadi va ular o'rtasidagi turlar va munosabatlar nuqtai nazaridan ifodalanadi. Shuningdek, "kontseptual sxema" atamasini bunday talqin qilishning o'ziga xos xususiyati shundaki, aniq vazifalarning kontseptual sxemalari ko'rib chiqiladi, ular uchun ularning maqsadlari va demak, ulardan kelib chiqadigan borliqni ko'rib chiqish kontekstlari aniq belgilanadi. Boshqa tomondan, "vazifa" atamasi bilan ifodalangan tushunchani ma'lum bir muhandislik muammosini belgilashdan ko'ra ancha kengroq talqin qilish mumkin. Vazifa juda murakkab bo'lishi mumkin va shuning uchun katta kichik vazifalar to'plamini o'z ichiga oladi.

Bunday holda, vazifa paradigmaga aylanadi. Paradigmaning maqsadlari ham murakkab va uning kontseptsiyasini tashkil qiladi. Ontologiyalarni rasmiy tavsiflashning foydaliligi aynan shunday tavsiflar bitta paradigma doirasida yaratilganligi va shuning uchun tavsiflangan ontologiyalar umumiy maqsadlarga ega bo'lib, ulardan birgalikda foydalanish mumkinligiga asoslanadi. Keling, kontseptual relativizm falsafasidan olingan muammoning kontseptual sxemasining rasmiy ta'rifini beraylik.

Ta'rif 2.1. Vazifaning kontseptual sxemasi - bu narsaning turlari va ular o'rtasidagi munosabatlar shaklida ifodalangan, berilgan vazifaning maqsadi va ushbu maqsad tomonidan qo'yilgan nuqtai nazar bilan belgilanadigan tajribani tashkil qilish usuli.

Yuqori darajadagi ontologiyalar

Ushbu bo'limda biz eng mashhur ontologiya kutubxonalarini ko'rib chiqamiz. Bunday kutubxonalar, qoida tariqasida, bir ontologiyada tavsiflangan tushunchalar boshqasida tavsiflangan tushunchalar asosida aniqlanganda, ierarxik munosabatlar asosida tashkil etiladi. Ushbu ierarxiyaning yuqori qismida yuqori darajadagi

ontologiyalar joylashgan. Yuqori darajadagi ontologiyalar o'zlarining o'ziga xos xususiyatlariga ko'ra, yuqori darajadagi mavhumlik tushunchalarini o'z ichiga olishi kerak, shuning uchun ular asosida turli xil muammolarning mavzu sohalarini kontseptsiyalash mumkin bo'ladi. Yuqorida biz har bir konseptual sxema ortida uning mazmunini belgilovchi aniq maqsad borligini aytdik. Yuqori darajadagi ontologiyalarning kontseptual sxemalarining mazmunini belgilaydigan maqsad faylasuflar tomonidan dunyoni ontologik tavsiflashda an'anaviy ravishda qo'yilgan maqsad bilan o'xshashdir - bu savolga javob: mavjudlik nima? Tarix shuni ko'rsatadiki, faylasuflar bu savolga turli yo'llar bilan javob berishgan, shuning uchun ularning izdoshlari bugungi kunda boshqalardan farq qiladigan yuqori darajadagi ontologiyalarning versiyalarini yaratishlari ajablanarli emas. Borliq tavsifining quyidagi versiyalarining har biri o'ziga xos afzallik va kamchiliklarga ega va, ehtimol, eng to'g'ri varianti mavjud emas, lekin tavsiflarning turli versiyalari bir-birini to'ldiradi va borliq haqida to'liqroq ko'rinish olish imkonini beradi. Shuni ham ta'kidlash kerakki, barcha tavsiflangan ontologik konstruksiyalar yagona kontseptual sxema doirasida yaratilgan bo'lib, uni Evropa sivilizatsiyasining madaniy matritsasi deb atash mumkin. Boshqa madaniyatlar, tarix ko'rsatganidek, boshqa ontologik mazmunga asoslanishi mumkin. Yevropa faylasuflarining ontologik qarashlari, asosan, Aristotel ontologik an'anasi izdoshlari bo'lgan qadimgi yunon mualliflaridan meros bo'lib qolgan. O'quvchi ko'rib turganidek, quyida keltirilgan yuqori darajadagi ontologiyalarning turli xil versiyalari asosan ushbu yondashuvga mos keladi. Aristotel qadimgi yunon tilining lingvistik tahliliga asoslanib, narsalarning o'nta asosiy toifasini belgilaydi. Aristotel ontologiyasidagi asosiy kategoriya barcha boshqa narsalarning birlamchi asosi sifatida substansiya kategoriyasidir. Miqdor va sifat kategoriyalari, agar ularning substansiyalari ana shu miqdor va sifatlarga ega bo'lsa, mavjud bo'ladi, narsalar o'rtasidagi munosabatlar kategoriyalari esa shu narsalarning substansiyalari o'zaro shu munosabatlar bilan bog'langandagina mavjud bo'ladi. Bu ma'noda narsalarning barcha boshqa toifalari moddalarga bog'liqdir. Shu nuqtai nazardan Aristotel Haqiqatni mustaqil mavjudotlar (moddalar) va qaram (boshqa barcha narsalar) ga ajratadi. Aristotel ontologiyasida yana bir shunday bo'linish - yuzaga keladigan va uzluksiz ob'ektlarga bo'linishdir.

Uzluksiz ob'ektlar o'z mavjudligining barcha davrida bir butun sifatida mavjud bo'lib, paydo bo'ladigan ob'ektlar esa, aksincha, o'zlarini bir butunga to'plash uchun vaqt oralig'iga muhtoj. Masalan, modda, miqdor, sifat va makon toifalarini uzluksiz mavjudotlar deb hisoblash mumkin, chunki ular vaqt oraliqlarini belgilashga muhtoj emas. Harakat, ishtiyoq (hayajon) va vaqtning o'zi kategoriyalari paydo bo'ladigan ob'ektlarga misoldir. Aristotel ham universal tushunchasini kiritadi. Aristotelning universali biz tushuncha yoki tip deb ataydigan narsa bilan sinonimdir. O'quvchi quyidagi ro'yxatdagi ontologiyalar Aristotel ontologiyasining yuqoridagi tavsifining turli xil variantlari ekanligiga ishonch hosil qiladi. Quyidagi ontologiyalar ro'yxati to'liq deb da'vo qilmaydi; uni taqdim etishdan maqsad, bir tomondan, o'quvchiga yuqori darajadagi ontologiyalar qurilgan tamoyillar haqida tushuncha berishdir, boshqa tomondan. qo'l, o'quvchini bugungi kunda eng mashhur ontologiya asoslari bilan tanishtirish. O'quvchi, shubhasiz, yuqori darajadagi ontologiyalarning falsafiy ontologiyalar bilan o'xshashligini ta'kidlaydi, ya'ni Dunyo metafizikasini tushuntirish uchun faylasuflar tomonidan qurilgan narsalar.

Jon Oulning "Olmosi"

Sun'iy intellekt bo'yicha amerikalik mutaxassis Jon Sova o'zining "Bilimning vakili" monografiyada yuqori darajadagi ontologiyaning o'z versiyasini maxsus turdagi (panjara) tartiblangan tuzilma sifatida taqdim etdi. Ushbu yuqori darajadagi kontseptual panjara "OWL olmosi" unvonini oldi. Jon Sova yuqori darajadagi ontologiyaning o'z versiyasini asosan Charlz Pirs va Alfred Uaytxedning falsafiy asarlariga asoslangan holda yaratdi. Yuqorida ta'kidlanganidek, "olmos" tushunchalari tartiblangan va bu tartib to'liq panjara tuzilishiga ega. OWL panjarasidagi tartib munosabati tasnifda ma'lum bo'lgan tip-kichik turdagi munosabatlardir. Panjaraning to'liqligi tartibli tuzilishdan yuqorida va pastda ikkita maxsus element mavjudligini aniqlaydi. "T" elementi "ob'ekt" atamasining sinonimi sifatida ko'rib chiqilishi mumkin, Owl panjarasining har qanday elementi ob'ektdir va shuning uchun bu elementning kichik turi. T bo'linmas bir butun sifatida qaraladigan eng mavhum ob'ekt sifatida ham qaralishi mumkin. Owl panjarasining pastki qismidagi element "⊥" bilan belgilanadi va "mumkin bo'lmagan mavjudot" - mantiqiy bo'lmagan absurd tushunchani ifodalaydi. Bu elementlar orasida

joylashgan tushunchalar real mazmunga ega. Owl ontologiyasi o'n ikki toifani o'z ichiga oladi, ular turli jihatlar bilan ajralib turadi. Yuqori darajada hodisalarni tavsiflovchi *uchta jihat mavjud*:

- **Birinchi jihat** ko'rib chiqishning berilgan elementi narsalarning real dunyosiga tegishlimi yoki hech qanday real ob'ektni bildirmaydigan mavhum tushunchami - bu mavzu sohasi elementlarini jismoniy va mavhumga bo'lish haqida gapiradi.

- **Ikkinchi jihat** elementlarning bir-biri bilan o'zaro bog'lanishiga tegishli. Agar element o'zi tomonidan ta'rifda boshqa elementlar bilan bog'lanishni talab qilmasdan aniqlangan bo'lsa, u mustaqil deb hisoblanadi. Agar element mavzu sohasining boshqa elementi bilan bog'liq holda aniqlansa, u nisbiy deb hisoblanadi. Nihoyat, agar element ikkita boshqa element o'rtasidagi bog'liqlik sifatida aniqlansa, u vositachi sifatida aniqlanadi. Bu jihatning har bir xususiyati, o'z navbatida, ikki toifaga bo'linishi mumkin. Mustaqil elementlarni tegishlilik va shakl jihatidan tavsiflash mumkin. Muvofiqlik haqiqatga bo'lgan munosabatni tavsiflaydi, ya'ni berilgan mavzu elementi ob'ektni yoki jarayonmi. Shakl ma'lum bir element qanday tavsiflanganligini, qandaydir sxema orqali statik yoki ba'zi bir skript orqali dinamik tarzda tavsiflanganligini aytadi. Tegishli elementlar bog'lanish usuli bilan tavsiflanishi mumkin: yoki element to'g'ridan-to'g'ri boshqa bir element tomonidan tushunishni ifodalaydi (keyin uni qo'shilish yoki ishtirok etish orqali olish mumkin) yoki element qandaydir tarzda boshqasi orqali gap shaklida ifodalanadi. (taklif). Ikkinchi holda, bu gap qandaydir tavsif (tavsif) yoki qandaydir tarix (vaqt bo'yicha tavsif - tarix) shaklida bo'lishi mumkin. Va nihoyat, agar element oraliq sifatida tavsiflansa, u jismoniy elementlar o'rtasidagi bog'liqlik yoki boshqa mavhum elementning niyati (niyati) bo'lishi mumkin.

- **Uchinchi jihat** berilgan elementning vaqt ichida qanday mavjudligini tavsiflaydi. Agar element etarlicha uzoq va barqaror vaqt davomida mavjud bo'lsa, unda bu elementni uzluksiz deb atash mumkin. Aks holda, buni hodisa deb ataymiz.

Har bir mavzu elementi quyidagi o'n ikkita toifadan biriga tegishli bo'lishi mumkin (toifa tavsiflari quyidagilardan olingan:

- **Ob'ekt.** Muayyan barqaror vaqt davomida mavjud bo'lishi mumkin bo'lgan mavjudot. Ob'ekt oddiy jismoniy ob'ekt bo'lishi

mumkin, shuningdek, ob'ektga yo'naltirilgan dasturlash tilida sinfning namunasi bo'lishi mumkin.

- **Jarayon.** Ko'rib chiqishning vaqt miqyosiga ko'ra, bir xil ob'ektlarni ob'ektlar va jarayonlar sifatida ko'rib chiqish mumkin. Masalan, olmos odatda ob'ekt sifatida tushuniladi, ammo agar biz etarlicha uzoq vaqtni, masalan, bir necha million yilni oladigan bo'lsak, olmosni uning atom tuzilishini o'zgartirish jarayoni deb hisoblash mumkin. Jarayonlar har doim o'zgarishlar sodir bo'ladigan boshlang'ich va yakuniy nuqtaga ega. Jarayonlar diskret va uzluksiz bo'linadi.

- **Sxema.** Diagramma mavhum shakl bo'lib, uning tuzilishi vaqt bilan bog'liq bo'lgan bog'lanishlarni o'z ichiga olmaydi. Tuzilishlarga misollar geometrik shakllar, tildagi jummalarning sintaktik tuzilmalari yoki raqamli fotosuratning mazmunini o'z ichiga oladi.

- **Skript.** Skript - vaqtlar ketma-ketligini ifodalovchi mavhum shakl. Skriptga misollar kompyuter dasturi, tort retsepti yoki musiqiy partiturasidir.

- **Splicing (Juncture).** Bog'lanish ikki yoki undan ortiq elementlar orasidagi bog'lanish sifatida tavsiflanishi mumkin, bu juda barqaror vaqt oralig'ida davom etadi. Bunga misol qilib, kamonda bir-birini qo'llab-quvvatlaydigan ikkita toshni keltirish mumkin. Ark - bir nechta shunday qo'shimchalardan tashkil topgan bog'ich.

- **Ishtirok etish.** Ishtirok etish ishtirokchini qiziqtiradigan vaqt oralig'ida sodir bo'ladi. Ishtirok etuvchi element ishtirokchi deb ataladi va ishtirok qandaydir jarayonda amalga oshiriladi.

- **Tavsif.** Tavsif - berilgan sxema uzluksiz mavjudotning ayrim tomonlarini qanday tavsiflashini tavsiflovchi jumla.

- **Tarix.** Tarix - bu skriptni ba'zi paydo bo'lgan shaxsning davrlari bilan bog'laydigan jumla. Masalan, kompyuter dasturi - bu skript, berilgan dasturni bajaruvchi kompyuterning o'zi jarayon (yangi ob'ekt), dastur ko'rsatmalari ketma-ketligida kodlangan axborot esa tarixdir.

- **Tuzilishi.** Struktura bir nechta elementlarni birlashtiradi va bu elementlar orasidagi bog'lanishlar strukturani tashkil qiladi. Shunday qilib, struktura qandaydir doimiy mavjudotning aloqasidir.

- **Vaziyat.** Vaziyat ishtirokchilarni muayyan jarayonda bog'laydi, uning turli davrlarida turli ishtirokchilar jalb qilinishi mumkin.

Shunday qilib, vaziyat - bu qandaydir paydo bo'lgan shaxsning aloqasi.

· **Sabab.** Oddiy tavsifdan farqli o'laroq, sabab xulosa sifatida taqdim etiladigan tushuntirishni o'z ichiga oladi. Misol uchun, tug'ilgan kunni nishonlash uchun tavsif sovg'alar ro'yxatini ko'rsatishi mumkin, ammo buning sababi ro'yxatning bayram bilan qanday bog'liqligini tavsiflashdir.

· **Maqsad.** Maqsad jarayonning mavjudligi sababini tushuntiradi. Masalan, "Tug'ilgan kuning bilan" qo'shig'ining so'zlari va partituralari ssenariyni tashkil qiladi, odamlarning tug'ilgan kunida qo'shiqni qanday kuylagani hikoyani tashkil qiladi, vaziyatni tushuntirish esa maqsaddir.

CYC

Cyc ("entsiklopediya" dan olingan va "Cyc" deb o'qiladi) - insonning kundalik tafakkurini taqlid qilish va shu tariqa mualliflarga bilimning turli sohalarida murakkab sun'iy intellekt muammolarini hal qilishda yordam berish uchun mo'ljallangan katta bilimlar bazasini yaratish loyihasi. Cyc loyihasining muallifi amerikalik AI tadqiqotchisi Duglas Lenat. Cyc bilimlari bazasi 1984 yilda o'sishni boshladi va shu vaqtdan beri o'sib bormoqda. Hozirda Cyc ma'lumotlar bazasida 2,2 millionga yaqin bayonot mavjud. Dastlab, Cyc loyihasining bilimlar bazasi ochiq foydalanish uchun mo'ljallanmagan, shuning uchun imtiyozsiz foydalanuvchilar to'plangan bilimlar zaxirasiga to'g'ridan-to'g'ri kirish imkoniga ega emaslar. Biroq, Cyc bilimlar bazasining ochiq qismi mavjud - OpenCyc deb ataladigan. Cyc loyihasining yaxshi sharhi M. Alekseevaning maqolasida keltirilgan. Cyc bazasidagi bilimlar maxsus tilda - CycLda "Har bir daraxt o'simlik" yoki "O'simliklar o'limli" kabi bayonotlar shaklida yozilgan. CycL sintaksisi LISP ga o'xshaydi, bayonotlar ham ro'yxat shaklida shakllanadi. Cyc shuningdek, bilimlar bazasidagi bayonotlar asosida savollarga javob beradigan xulosa chiqarish mexanizmiga ega. Cyc bilimlar bazasida umumiy yuqori darajadagi tushunchalarni belgilaydigan faktlarni ajratib ko'rsatish mumkin. Bu faktlar yuqori darajadagi ontologiyani ifodalaydi. Biz Cyc loyihasi tizimini 2-bobda batafsil ko'rib chiqamiz. Shuning uchun bu yerda biz Cyc loyihasining yuqori darajadagi ontologiyasi mazmunini batafsil tavsiflab bermaymiz.

Asosiy rasmiy ontologiya (BFO)

BFO loyihasi ontologik modellashtirish sohasidagi mashhur amerikalik tadqiqotchi Barri Smit rahbarligida ishlab chiqilgan. Cycdan farqli o'laroq, BFO loyihasi dastlab turli ilmiy loyihalarda foydalanish uchun mo'ljallangan yuqori darajadagi ontologiya sifatida yaratilgan. Xususan, BFO OBO Foundry (Ochiq tibbiy ontologiya) loyihasida qo'llaniladi. BFO ontologiyalari vaqt ichida mavjud bo'lishning turli usullariga yoki fazo-vaqt o'zaro ta'sirining ikkita asosiy rejimiga mos keladigan ikki turga bo'linadi.

1. Birinchi tur vaqt tushunchasini o'z ichiga oladi, ya'ni bunday ontologiyalarda vaqt allaqachon qandaydir tarzda aniqlangan deb taxmin qilinadi va berilgan ontologiyada hisobga olinmaydi. Aytishimiz mumkinki, ikkinchi turdagi ontologiyalarda to'rt o'lchovli ob'ektlardan (3D + Vaqt) farqli o'laroq, faqat uch o'lchovli (3D) ob'ektlar ko'rib chiqiladi. Birinchi turdagi ontologiyalar haqiqatning bir lazasini aks ettiruvchi fotosuratlariga o'xshaydi va haqiqatni bunday fotosuratlar ketma-ketligi deb hisoblash mumkin. Ushbu turdagi ontologiyalar SNAP - ontologiyalar (inglizcha snapshot - snapshot) deb ataladi.

2. Ikkinchi tur ob'ektlarni vaqtida ko'rib chiqadi, ya'ni fazo-vaqt shaklining to'rt o'lchovli uzluksizligida. muhandislik nuqtai nazaridan, bunday ontologiya vaqtni qamrab oladi, shuning uchun bunday turdagi ontologiyalar SPAN - ontologiyalar deb ataladi. Shunday qilib, BFO predmet sohasini ikkita asosiy nuqtai nazardan tavsiflashga imkon beradi: ko'rib chiqilayotgan ob'ektning shakli va statik sxemasini tavsiflovchi fazoviy va mavzu sohasining dinamik xususiyatlarini aks ettiruvchi vaqtinchalik. Ta'riflar bir-birini to'ldiradi va birgalikda hayotning to'liq tavsifini beradi. SNAP - ontologiyalar ko'rib chiqish uchun etarlicha uzoq vaqt davomida o'zgarishlarga chidamli bo'lgan ob'ektlarni tavsiflaydi, SPAN - ontologiyalar jarayonlarni tavsiflaydi. Shunday qilib, barcha ko'rib chiqiladigan ob'ektlarni doimiy va o'zgaruvchanlarga bo'lish mumkin. Doimiy ob'ekt o'lchovli va mustaqil yoki qaram ob'ekt bo'lishi mumkin. O'zgaruvchan ob'ekt, shuningdek, o'lchov, vaqtinchalik koordinata bilan tavsiflanadi, dinamik xususiyatga ega (bu jarayon, jarayonning ajralmas qismi yoki jarayonlar to'plami bo'lishi mumkin), shuningdek, uning fazoviy tarkibiy qismlariga munosabati bilan tavsiflanadi: u

kosmosda harakatsiz yoki harakatlanuvchi narsalarni tavsiflaydi. BFO ontologiyasi quyidagi tillarda amalga oshiriladi:

- **OWL** - bu Internetning semantikasini tavsiflash uchun mo'ljallangan tavsiflovchi mantiqqa asoslangan til. Quyida biz ushbu til haqida batafsil gaplashamiz.
- **Isabelle** - birinchi tartibli mantiq tili, amerikalik ontolog Tomas Bitner tomonidan yaratilgan.

DOLCE

DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering - Descriptive Ontology for Linguistic and Cognitive Engineering) WonderWeb Foundational Ontologies Library deb ataladigan modullarning birinchisi - "WonderWeb" deb nomlangan yuqori darajadagi ontologiya kutubxonasi. Ushbu kutubxona ontologiyalarining asosiy maqsadi turli xil agentlarning o'zaro ta'sirini qo'llab-quvvatlashdir, bu ko'pincha AI maqsadlari uchun ishlab chiqilgan kompyuter dasturlari. Aniqroq aytganda, WonderWeb loyiha kutubxonasining ontologiyalari quyidagi maqsadlarda ishlab chiqilgan:

- Yangi ontologiyalarni rivojlantirish uchun boshlang'ich nuqta bo'ling. Ontologiyani ishlab chiquvchining eng qiyin vazifalaridan biri bu muammoni hal qilish uchun zarur bo'lgan tushunchalarni mavzu doirasidan tanlashdir. WonderWeb ontologiyalari muammoning asosiy yuqori darajadagi ko'rinishini taqdim etish orqali yordam berishi mumkin, bu esa berilgan muammoning ontologiyasini qurish uchun konkretlashtirilishi mumkin.
- Turli ontologiyalarni qat'iy taqqoslash uchun vositachi bo'ling.
- Mavjud ontologiyalarni birlashtirish uchun asos bo'ling.

WonderWeb amaliy ontologiya laboratoriyasida (LOA — Laboratory for Applied Ontology) taniqli italiyalik mutaxassis Nikolo Guarino rahbarligida ishlab chiqilmoqda. DOLCE WonderWeb loyiha kutubxonasining modullaridan birini ifodalaydi, tabiiy til ontologiyasi va uning kognitiv ma'nolari asosidagi tushunchalarni rasmiylashtirish uchun mo'ljallangan. DOLCE mualliflari, masalan, WordNet lingvistik ontologiyasining mazmunini aniqlashtirish uchun ushbu ontologiyadan foydalanishni taklif qilishadi. Shunday qilib, DOLCE ontologiyasi uning mazmunini belgilaydigan aniq pragmatik maqsadga ega - bu OWL olmosi yoki yuqorida tavsiflangan BFO

loyihasidagi kabi voqelikning umumiy ontologik toifalari to'plami emas, balki tilshunoslar va kognitiv olimlar uchun yaratilgan ontologiya. Loyiha mualliflari DOLCE ontologiyasi mazmuni kognitiv va lingvistik kontekstlarga ega ekanligini bildirish uchun "kognitiv tarafkashlik" atamasidan foydalanadilar. Agar OWLning "olmosi" faqat umumiy mavhum toifalarni tavsiflagan bo'lsa, DOLCE ontologiyasi, aksincha, tafsilotlar to'plamidir. Bu yerda xususiylar umumiy tushunchalardan (universallardan) farqli o'laroq, nusxalari bo'lmagan tushunchalarni bildiradi. DOLCE ontologiyasi multiplikativ deb ataladigan yondashuvga asoslanadi, bu holda ontologiyaning ba'zi ob'ektlari bir xil fazo-vaqt nuqtasida joylashgan bo'lishi mumkinligini anglatadi. Masalan, vaza va loy bo'lagi o'rtasidagi farq nima? Asosiy farq shundaki, "vaza" tushunchasi uchun uning shakli va topologik xususiyatlari o'zgarmasdir, bu "loy bo'lagi" tushunchasiga to'g'ri kelmaydi. Shuning uchun biz bu tushunchalarni aniqlab emas, balki vaza loy bo'lagidan iborat deb faraz qilgan holda, bu ob'ektni ham vaza, ham loy bo'lagi deb atashimiz mumkin.

DOLCE ontologiyasining barcha ob'ektlari quyidagi to'rtta xususiyatga ko'ra bo'linadi (ikki dixotomiyaga bo'lingan tetraxotomiya: barqaror va beqaror, sifatlar va hududlar):

- **Chidamli ob'ektlar.** Chidamlilik OWL olmosida uzoq umr ko'rish bilan bir xil degan ma'noni anglatadi. Bardoshli ob'ektlar ular mavjud bo'lgan barcha vaqtda to'liq mavjud.

- **Doimiy ob'ektlar.** beqaror ob'ektlar ularni ko'rib chiqish kontekstida paydo bo'ladigan ob'ektlar bo'lib, ularning tarkibiy qismlari ham o'zgarishi mumkin. BFO terminologiyasida bu vaqtning turli nuqtalarida olingan beqaror ob'ektlarning suratlari bir-biridan farq qilishi mumkinligini anglatadi, ammo bu barqaror ob'ektlar (SNAP va SPAN ob'ektlari) uchun emas.

- **Sifat.** Sifatlarni idrok etish yoki o'lchash mumkin bo'lgan asosiy xususiyatlar sifatida ko'rish mumkin. Sifatlariga quyidagilar kiradi: ranglar, o'lchamlar, tovushlar, hidlar, og'irliklar, uzunliklar, elektr zaryadlari va boshqalar. Sifatni " Property " atamasining sinonimi sifatida ko'rish mumkin, ammo Property toifalar-universallarga tegishli, sifat esa o'ziga xosdir. Sifatlar turlarga bo'linadi, ya'ni hidlar hid sifatining bir turi, og'irliklar tarozi va

- **Mintaqalar.** Mintaqalar Gardenfors kontseptual maydonining doimiy subdomenlaridir. DOLCE ontologiyasining mohiyati sifatida

barcha fazilatlar har xil va individualdir, ammo ular berilgan ontologiyaning kontseptual makonining bir xil nuqtalariga murojaat qilishlari mumkin.

Biz bu yerda bo'sh joy etishmasligi va asosiy taqdimotdan chetga chiqmaslik uchun DOLCE ontologiyasining keyingi taksonomiyasini tavsiflamaymiz. DOLCE ontologiyasi OWL tilida amalga oshiriladi va uni Amaliy Ontologiya Laboratoriyasi veb-saytidan bepul yuklab olish mumkin.

Umumiy rasmiy ontologiya (GFO)

GFO dastlab yuqori darajadagi ontologiya sifatida yaratilgan bo'lib, undan turli maqsadlarda kontseptual modellashtirish amalga oshirilishi mumkin. GFO ontologiyasi Onto-Med tadqiqot guruhi a'zolari tomonidan ontologik modellashtirish bo'yicha mashhur nemis mutaxassisi Geynrix Gerre rahbarligida ishlab chiqilgan. Masalan, aniq lingvistik va kognitiv kontekstga ega bo'lgan DOLCE ontologiyasidan farqli o'laroq, GFO so'nggi davrlarning falsafiy nazariyalariga muvofiq haqiqatni tavsiflash uchun yaratilgan. Ontologiyaning elementlari sinflar, to'plamlar va atomlardir. Sinflar va to'plamlar odatiy to'plam-nazariy ma'noda farqlanadi, ya'ni dunyo turlarga bo'lingan, bu yerda atomlar va to'plamlar 0 tipidagi elementlardir va shuningdek, 0 tipidagi barcha elementlarning $C [0]$ sinfi mavjud. Mavjud sinflarga asoslanib, siz "bularning barcha elementlari" yangi sinfini qurishingiz mumkin. sinflar". Bu ajralish Rassellning paradoksi bilan bog'liq qiyinchiliklardan qochadi. Rasmiy munosabatlarning maxsus sinfi ham mavjud. Atomlar individual, universal yoki fazo-vaqt ob'ektlari bo'lishi mumkin. Individ - makon va vaqtda mavjud bo'lgan yagona shaxs. Universallarning nusxalari bor. Individuallar universallarning misollari sifatida qabul qilinadi, ya'ni shaxslarda mavjud, lekin ularsiz mavjud bo'lolmaydi. Bu pozitsiya, shubhasiz, Aristotelning universal tushunchasiga bo'lgan yondashuviga to'liq mos keladi. Universallarni tafakkurning asosiy birliklari sifatida ham tushunish mumkin. Atomlarning fazo-vaqt xossalari xronoid va topoid toifalari orqali ifodalanadi. Xronoidlar deganda o'zaro bog'liq vaqt oralig'i, topoidlarda, mereotopologik tuzilishga ega bo'lgan fazoviy mintaqalar tushuniladi (ya'ni, ularning qismlaridan ma'lum tarzda tuzilgan). Yuqorida aytib o'tganimizdek, individlar fazo va vaqt ichida singdirilgan atomlardir. Ushbu amalga oshirish individlarning ob'ektlar va jarayonlarga bo'linishi bilan

tavsiflanadi. Biz bu farqni yuqorida muhokama qildik (BFO ontologiyasidagi SNAP va SPAN ob'ektlari, shuningdek DOLCE ontologiyasining barqaror va beqaror elementlari). GFO shuningdek, modda tushunchasidan foydalanadi, u xususiyatga ega bo'lgan va kosmosda kengaytmaga ega bo'lgan ob'ekt sifatida tushuniladi. Boshqa turdagi ob'ektlar - bu momentlar, faqat boshqa ob'ektlarning bir qismi sifatida mavjud bo'lgan ob'ektlar (masalan, elektr kuchlanish faqat ma'lum bir o'tkazgichda mavjud). Barcha jarayonlar vaqtinchalik tushunchasi bilan tavsiflanadi va aslida jarayonlar, harakatlar, o'zgarishlar, tarixlar, holatlar va jarayonlar chegaralariga bo'linadi. Moddalar o'z momentlari bilan birgalikda konfiguratsiyalarni hosil qiladi. Bir butun sifatida ko'rilgan kompozit konfiguratsiya vaziyat deb ataladi. Vaziyat tushunchasi Jon Barvis va Jon Perrining vaziyatlarning matematik nazariyasini tavsiflashga bag'ishlangan ishlariga asoslanadi. GFO ontologiyasida qo'llaniladigan vaziyatning ta'rifi vaziyat nazariyasida qo'llaniladiganidan farq qiladi, lekin umumiy ma'noda u ikkinchisiga o'xshaydi. GFO vaziyat tushunchasining ikki tomonlama ta'rifidan foydalanadi: statik (vaziyat) va dinamik (situoid). Situoid - bu butun dunyoning bir qismi, ammo bu qismni tashkil etuvchi barcha ontologik mavjudotlarning to'liq tarixi bilan birga. Shunday qilib, har bir situoid vaqt bo'yicha davomiylikka ega va uning fazoviy konfiguratsiyasini ifodalovchi ba'zi topoidlarga o'ralgan. Vaziyat - situoidning ma'lum bir vaqtning o'zida atomik sifatida ko'rilgan statik surati. Misol uchun, "Ivan Masha" iborasini ko'rib chiqing. Bu yerda Ivan va Masha bir-birlari bilan "ko'rishish" munosabati bilan bog'langan, ammo bu konfiguratsiya hali situoid hosil qilmaydi. Situoidning shakllanishi uchun ushbu iborada tavsiflangan jarayon sodir bo'lgan muhitni hisobga olish kerak. Ehtimol, Ivan Masha bilan o'rmonda yurgandir yoki ular gavjum ko'chada skameykada o'tirishgandir. "Ivan Masha" iborasi bilan tavsiflangan vaziyatning paydo bo'lish tarixining to'liq tavsifi, bu sodir bo'lgan joyning tavsifi bilan birgalikda situoidni hosil qiladi. Monografiyada GFO ontologiyasining tuzilishi haqida yaxshi ma'lumot berilgan.

IDEAS

IDEAS (International Defense Enterprise Architecture Specification) – ma'lumotlar almashinuvi uchun xalqaro mudofaa sanoati me'morchiligi spetsifikatsiyasi. Birjani standartlashtirish

uchun IDEAS deb ham ataladigan ontologiyadan foydalaniladi. IDEAS ontologiyasi yuqori darajadagi ontologiyaga, IDEAS Foundation va aniqroq IDEAS modeliga bo'linadi. IDEAS ontologiyasi ontologiyaning barcha ob'ektlari uchta fazoviy va bitta vaqtinchalik o'lchovdan iborat 4D kontinuumda joylashgan deb taxmin qilinadigan yondashuvdan foydalanadi. IDEAS fondi sanoat ontologiyalarini ishlab chiqish uchun BORO deb ataladigan usul yordamida ishlab chiqilgan bo'lib, uning mazmuni ushbu bobda keyinroq muhokama qilinadi. IDEAS Foundation ontologiyasi juda oddiy tuzilishga ega va to'rtta asosiy toifaning elementlarini o'z ichiga oladi:

- Individlar jismoniy kengaytmaga ega bo'lgan har qanday narsadir.
- Turlar — individlar, turlar va kortejlar to'plami.
- Kortejlar ikki yoki undan ortiq individlar, tiplar yoki kortejlar o'rtasidagi munosabatlardir.
- Ob'ektlar - ontologiyaning ildiz kategoriyasi, individual, tip yoki kortej bo'lishi mumkin.

Jismoniy shaxslarga misollar: Sen, men, mening mashinam, Yer, Ostankino teleminorasi. Turlarga misollar: sariq narsa, odamlar, binolar, darajalar (darajali - elementlari turlar bo'lgan tur). Tuplarga misollar: mening mashinam va uning dvigateli (yaxlit), sariq narsalar va Nyu-York taksilari (turdagi kichik tip), qora narsalar va mening mashinam (nasl). IDEAS Foundation RDFda kodlanganligi sababli, undagi munosabatlarning katta qismi ikkilik munosabatlardir. Bu haqiqat ontologiyada bug 'toifasi bo'yicha o'z aksini topadi. IDEAS Foundation ontologiyasi uchta oldindan belgilangan munosabatlarga ega:

- **Qisman-butun.** Bu munosabat ikki individni birlashtiradi, ulardan biri bir butun, ikkinchisi esa uning bir qismidir. IDEAS ontologiyasi 4D fazo-vaqt uzluksizligini modellashini hisobga olgan holda, bu munosabat ba'zi bir jarayonni (vaqt oralig'ini) uning ba'zi bajarilish bosqichlari bilan ham bog'lashi mumkin.
- **Subtip turi.** Turni boshqa tur bilan bog'laydi. Kichik turning bir nechta nusxalari supertipning bir nechta misollari ichida joylashtirilgan deb taxmin qilinadi.
- **Namuna turi.** Turni o'zi ifodalagan to'plam elementlari bilan bog'laydi.

WordNet

WordNet - bu semantik lug'at deb ataladigan shaklda taqdim etilgan ingliz tilining tavsifi bo'lib, u semantik tarmoqning ma'lum bir turida o'rnatilgan uning elementlari o'rtasidagi semantik munosabatlar bilan to'ldirilgan oddiy lug'atdir. Ingliz tilidagi so'zlar to'plami sinonimiyaga nisbatan omillashtiriladi, bu omillashtirish sinfi sinset deb ataladi. Sinsetlar o'rtasida semantik munosabatlar o'rnatiladi. WordNet-ning dizayn maqsadi ikki xil: bir tomondan, intuitiv ingliz lug'atini taqdim etish va boshqa tomondan, ingliz tilida so'zlarni avtomatlashtirilgan qayta ishlash va boshqa AI faoliyati uchun ishlatilishi mumkin bo'lgan vositani taqdim etish. WordNet ma'lumotlar bazasida taxminan 115 ming sinsetga guruhlangan 150 mingga yaqin so'z ta'riflari mavjud va siqilgan shaklda taxminan 12 gigabaytni tashkil qiladi. WordNetning rivojlanishi 1985 yilda Prinston universitetining Kognitiv ilmiy laboratoriyasida falsafa professori Jorj Miller rahbarligida boshlangan. WordNet-ni leksik ontologiyaning alohida turi sifatida ko'rish mumkin. Biroq, bunday taqdimot qo'shimcha ishni talab qiladi. WordNet bazasining har bir sinsetini semantik birlik deb hisoblash mumkin, ya'ni ontologik kategoriya. WordNet-da sinsetlar o'rtasida giponimik munosabat mavjud bo'lib, uning asosida taksonomiya tuzilishi mumkin, bunda har bir giponim o'z giperonimining bolasi sifatida ifodalanadi. Biroq, ba'zi juftliklar uchun giponimiya bilan bog'liq holda, taksonomik juftlikni (tur-kichik tip munosabatlari juftligini) emas, balki bir juft tip-nasl munosabatlarini shakllantirish kerak. Shunday qilib, WordNet ma'lumotlar bazasini ontologiyaga aylantirish qo'shimcha ishlarni talab qiladi. Bunday ish olib borildi, natijada olingan ontologiya WebKB-2 deb nomlanadi, uning tavsifini topish mumkin. Shunday qilib, WordNet-ni lingvistik maqsadlarda yaratilgan yuqori darajadagi ontologiya deb hisoblash mumkin, agar siz bunday ontologiyaga WordNet ma'lumotlar bazasidagi eng umumiy toifalarni kiritsangiz.

Tavsiya etilgan yuqori birlashtirilgan ontologiya (SUMO)

SUMO - bu IEEE Standard Upper Ontology Working Group tomonidan ishlab chiqilgan yuqori darajadagi ontologiya. IEEE ishchi guruhining maqsadi ma'lumotlar almashinuvi, ma'lumotlarni qidirish, avtomatik xulosa chiqarish va tabiiy tillarni avtomatlashtirilgan qayta ishlash kabi kompyuter injiniringi sohalariga mos keladigan ontologiyani ishlab chiqish edi. Shu sababli, SUMO bir nechta taniqli

ontologiyalarning birlashmasi bo'lib, Ontolingua serverida joylashgan ontologiyalar, Jon Owlning yuqori darajadagi ontologiyasi (OWLning "olmosi") va boshqalar. SUMO ontologiyasi IEEE ishchi guruhi maqsadlari uchun moslashtirilgan KIF tili - SUO-KIF versiyasida ishlab chiqilgan.

SUMO ontologiyasi quyidagi tuzilishga ega:

- Ontologiyaning ildizi kategoriya borligidir.
- Ob'ekt mavhum yoki jismoniy bo'lishi mumkin.
- Jismoniy shaxslar ob'ektlar yoki jarayonlardir.
- Abstrakt ob'ekt - bu miqdor, atribut, sinf (to'plam), munosabat, bayonot, grafik yoki grafik elementi.
- Ob'ekt odamlar odatda ushbu tushuncha bilan nimani nazarda tutayotganini ifodalaydi. Ob'ektlarga misollar umumiy jismoniy ob'ektlar, geografik hududlar va jarayonlarning joylashuvi.
- Jarayon deganda ma'lum bir joyda, ma'lum bir vaqtda sodir bo'ladigan narsa tushuniladi. Jarayon ikki ob'ekli jarayon, maqsadli jarayon, harakat, ichki o'zgarish yoki topologik o'zgarishdir.
- Miqdor - bu narsaning nechta elementi borligi yoki uning qanchalik kattaligining har qanday spetsifikatsiyasi. Shunga ko'ra, ushbu toifaning ikkita kichik sinfi mavjud: son va jismoniy miqdor.
- Atribut - kategoriya ob'ektining pastki sinfi hisoblanmaydigan sifat.
- Sinf mavhum ob'ekt bo'lib, unda elementlar yoki misollar bo'lishi mumkin.
- Aloqa - bu kortejlar to'plamini ifodalovchi sinfning maxsus turi. Uch xil aloqa mavjud: predikatlar, funktsiyalar va ro'yxatlar. Ro'yxat qisman tartiblangan to'plamdir.
- Gap to'liq fikrni yoki shunday fikrlar majmuini ifodalovchi mavhum borliqdir. Masalan, formula (masalan, Vaska Kot) Vaska "Kot" to'plamining elementi ekanligini aytadi.
- Grafik o'zaro bog'liq sinflar juftligi: GraphNodes tugun sinfi va GraphArcs chekka sinfi.
- Grafik elementi grafikning bo'linmas qismidir, ya'ni. yoki tugun yoki chekka.

SUMO ontologiyasi juda katta, shuning uchun biz bu yerda uning barcha toifalarini bermaymiz, chunki yuqoridagi ro'yxat

o'quvchi ushbu ontologiya nima ekanligini tushunishi uchun yetarli bo'ladi. Aytaylik, bu ontologiyada shaxs bilvosita ajdod sifatida agent toifasiga ega bo'lgan kognitiv agent toifasining misoli sifatida qaraladi. SUMO ontologiyasining qiziqarli xususiyati shundaki, u uchun WordNet bazasining ontologiyasi bilan to'liq yozishmalar yaratilgan. Shunday qilib, SUMO amalga oshirilayotgan WordNet bilan bog'liq turli xil lingvistik ishlanmalarda qo'llanilishi mumkin: lingvistik mavzularda juda ko'p asarlar mavjud.

Ontologiyani qurish metodologiyasi

Ushbu bo'limda biz ontologiyani rivojlantirishning mashhur usullarini ko'rib chiqamiz. Muallif modellashtirish jarayonida duch kelishi kerak bo'lgan turli xil muammolarni kengroq ko'rsatadigan misollarni tanlashga harakat qildi. Biz quyidagi uchta metodologiyani ko'rib chiqamiz:

1. UML tilidan foydalangan holda dasturiy ta'minot tizimlarini modellashtirish.
2. Tomas Gruber tomonidan amaliy ontologiyalarni ishlab chiqish metodikasi.
3. Mavjud dasturiy ta'minot tizimlarini qayta tashkil etishning BORO usuli.

UML yordamida modellashtirish

Dasturchilar orasida ishlab chiquvchilar rollarini qat'iy belgilashga asoslangan ishlab chiqish usuli juda mashhur bo'ldi. Dasturni qurish ikki asosiy bosqichga bo'linishi mumkin:

1. Dastur vazifasining kontseptual sxemasini tushunish bosqichi — **loyihalash**. Bunga ushbu dasturiy ta'minot tizimi mo'ljallangan mavzu bo'yicha mutaxassislar bilan muloqot qilish va ushbu muammoni hal qilish uchun mos keladigan adekvat muhandislik modelini yaratish kiradi. Agar dasturiy ta'minot loyihasi nisbatan sodda bo'lsa, ya'ni dasturiy ta'minot tizimini bir nechta izolyatsiyalangan modullarga bo'lishni talab qilmaydi, ularning har biri o'z kichik vazifani hal qiladi, keyin odatda muammo sohasining kontseptual diagrammasi rasmiylashtirilmaydi, dastur kodi yoki muammoni hal qilish algoritmining blok diagrammasi darhol yoziladi. Ammo, agar dasturiy ta'minot tizimi katta bo'lsa, unda bunday rasmiylashtirish mutlaqo zarurdir. Dasturiy ta'minot tizimi uchun uning turli modullari bir-biri bilan qanday o'zaro ta'sir qilishini qat'iy belgilash juda muhim, ya'ni o'zaro aloqa protokolini ishlab chiqish.

Buning uchun, birinchi navbatda, berilgan topshiriqning dasturiy modelini qat'iy tavsiflash kerak, bu uning konseptual sxemasidan boshqa narsa emas. Bunday rasmiy tavsiflarni qurish uchun tegishli mutaxassislar kerak, shuning uchun dasturlash muhandisligida yangi mutaxassislik - dasturiy ta'minot dizayneri yoki dasturiy ta'minot tizimi arxitektori paydo bo'ldi.

2. **Kodlash.** Ongli kontseptual sxemaga asoslanib, siz dastur kodini yozishingiz mumkin. Bu, odatda, tegishli mutaxassislar - kodlovchilar tomonidan amalga oshiriladi. Aytishimiz mumkinki, koder - bu dasturchi, uning vazifasi dizayner tomonidan taqdim etilgan dastur modeli spetsifikatsiyasiga muvofiq ma'lum bir dasturlash tilida dastur kodini yozishdir. "Sof" koderning ishi, albatta, zerikarli va ijodiy emas, shuning uchun ko'pincha koderlar ma'lum bir dizayn vazifalarini ham bajaradilar, masalan, ma'lum bir dasturlash tili bilan bog'liq. Bunday vazifalarga, masalan, turlar ierarxiyasini ishlab chiqish va ular orasidagi interfeysni spetsifikatsiya qilish kiradi. Shunday qilib, bizning nuqtai nazarimizdan, loyihalash bosqichi muammoning ontologiyasini qurishdan boshqa narsa emas, uni hal qilish ushbu dasturiy ta'minot tizimiga bag'ishlangan. Rivojlanishning ushbu bosqichi uchun murakkab vositalar mavjud. Shuningdek, dasturlash modeli spetsifikatsiyasi tillari ixtiro qilindi. Biz shunday tillardan birini ko'rib chiqamiz - UML.

Unified Modeling Language (UML) - bu dasturiy ta'minot modellarini loyihalash uchun maxsus mo'ljallangan grafik til. UML - bu grafik til, ya'ni tilning elementlari turli xil tavsiflar - grafik piktogrammalarning matn bilan kombinatsiyasi va dastur modellarining tavsifi diagrammalar, UML tilining elementlaridan tuzilgan kombinatsiyalangan diagrammalar yordamida amalga oshiriladi. . Yagona modellashtirish tili Rational Software kompaniyasida 1995 yilda uning ikki hamkori Gredi Butch va Jeyms Rambeau tomonidan yaratilgan. UML ikki til asosida yaratilgan: dasturiy ta'minotni loyihalash tili Booch (Grady Butch) va dasturiy tahlil tili OMT (Object Modeling Technique) (Jeyms Rambeau). 1997 yil yanvar oyida Microsoft va Oracle korporatsiyasi kabi taniqli korporatsiyalar bo'lgan ko'plab kompaniyalar o'rtasidagi hamkorlik natijasida yaratilgan UML 1.0 spetsifikatsiyasi chiqarildi. UML tezda mashhur bo'ldi va deyarli 10 yil davomida o'ziga xos yagona keng tarqalgan til bo'lib qoldi. Bu yerda UML ning qisqacha tavsifi.

UMLda dasturlash modellarini belgilash uchun uch turdagi diagrammalardan foydalaniladi: Struktura diagrammasi dasturlash modelining strukturasi belgilash uchun ishlatiladi. Strukturaviy diagrammalar quriladigan asosiy element bu sinfdir. Sinf - bu atributlar (xususiyatlar) va operatsiyalar (usullar) yig'indisidir. Aslida, sinf - bu xususiyatlar to'plami, ya'ni Aristotel an'anasidagi tushuncha. Amaliyotlar UML terminologiyasida ob'ektlar deb ataladigan sinf misollari bilan qanday munosabatda bo'lishni aniqlaydi. Har bir sinf, shuningdek, nima uchun zarur bo'lganligi - javobgarlik deb ataladigan tavsif bilan ham bog'lanishi mumkin. Har bir sinf, shuningdek, sinflarni bir-biridan ajratish uchun ishlatiladigan o'ziga xos nomga ega. Sinf to'rt qismga bo'lingan to'rtburchaklar orqali tavsiflangan. Yuqori qismida sinf nomi yoziladi, ikkinchi qismida yuqoridan uning atributlari (har bir atribut uchun bitta qator ajratiladi), uchinchi qismda sinf bilan bajariladigan amallar, to'rtinchi qismida esa uning atributlari ko'rsatilgan. qismida uning majburiyatlari tavsifi mavjud. Sinflar o'rtasida munosabatlar o'rnatilishi mumkin. UMLda aloqalarning uch turi mavjud:

- **Bog'liqlik munosabatlari.** Bog'liqlik - bu bir sinfning spetsifikatsiyasining o'zgarishi boshqasining xatti-harakatiga ta'sir qilishi mumkin bo'lgan foydalanish munosabati. Misol uchun, isitish quvurlari ular orqali oqadigan suvni isitish uchun isitgichga bog'liq.
- **Umumlashtirish munosabati.** Bu munosabat kichik turni uning asosiy turi bilan bog'laydi. UML mos ravishda turlar va ularning kichik turlari uchun superklass va subklass atamalaridan foydalanadi. Masalan, "to'rtburchak" 90 graduslik burchakli to'rtburchaklar, "romb" esa tomonlari teng bo'lgan to'rtburchaklardir. To'rtburchaklar va olmos sinflari to'rtburchaklar yuqori sinfining pastki sinflaridir.
- **Uyushma munosabatlari.** Bu munosabatlar ob'ektlar orasidagi tizimli munosabatlarni ifodalaydi: xonalar eshik va derazalarni o'rnatish mumkin bo'lgan devorlardan iborat.

Yuqoridagi munosabatlar turlarining har biri UMLda o'ziga xos kichik tiplarga ega, ammo bu yerda biz o'quvchini asosiy voqeadan chalg'itmaslik uchun uni batafsil ko'rib chiqmaymiz. Sinflar diagrammalarning maxsus turida - sinf diagrammalarida ko'rsatilgan. Sinf diagrammalarida sinflarning tavsiflari va ular o'rtasidagi

munosabatlar mavjud. UML shuningdek, sinflar misollarini va ular o'rtasidagi munosabatlarni bir vaqtning o'zida tavsiflaydigan ob'ekt diagrammalaridan foydalanadi. Ob'ekt to'rtburchak bilan tavsiflangan, unda avval ob'ektning nomi, so'ngra ikki nuqta bilan ajratilgan holda, ushbu ob'ekt namunasi bo'lgan sinf nomi yoziladi. Sinf atributlarining o'ziga xos qiymatlari gorizontol chiziq ostida keltirilgan. UMLdagi sinflar paketlangan bo'lishi mumkin, bu modullikning boshqa darajasini belgilaydi. Paket to'rtburchak shaklida tavsiflangan, yuqori chap tomonida paket nomi yopishtirilgan kichik to'rtburchaklar. Paket klassi nomlari katta to'rtburchak ichida keltirilgan. Paketlardagi sinflarga tashqaridan kirish mumkin, keyin sinf nomi oldiga plus qo'yiladi yoki mavjud emas, keyin esa sinf nomi oldiga minus qo'yiladi. Sinf atributlariga tashqi kirish ham ko'rsatilgan.

Paketlar paket diagrammalarida ko'rsatilgan. Barcha turdagi struktura diagrammalari quyida keltirilgan:

- *Sinflar (sinf).*
- *Komponentlar.*
- *Kompozit tuzilma. Hamkorlik (UML 2.0)*
- *Joylashtirish.*
- *Ob'ektlar (Ob'ekt).*
- *Paket.*

Xulq-atvor sxemalari. Xulq-atvor diagrammalarining uch turi mavjud:

- **Faoliyat (Faoliyat).** Buning diagrammalarida ayrim faoliyatning uning tarkibiy elementlariga parchalanishi ko'rsatilgan. Faoliyat deganda bo'ysunuvchi elementlarning o'zaro bog'langan ketma-ket va parallel bajarilishi ko'rinishidagi bajariladigan xatti-harakatlarning spetsifikatsiyasi tushuniladi - ichki harakatlar va bir tugunning chiqishidan boshqasining kirishiga o'tadigan iplar bilan bog'langan alohida harakatlar (harakatlar). Faoliyat diagrammasi biznes jarayonlarini, texnologik jarayonlarni modellashtirishda, ketma-ket va parallel hisoblashda qo'llaniladi. Faoliyat diagrammalari holat diagrammalariga juda o'xshaydi, agar davlatlar deganda biz individual harakatlarni nazarda tutsak va bajarilish iplari holatlar o'rtasidagi o'tish sifatida qabul qilinsa, faqat ikkinchisida harakatlarning parallel bajarilishini modellashtirish imkoniyati yo'q. Shuningdek, faoliyat diagrammalarida siz bajarilish iplarini treklarga

bo'lishingiz mumkin, ularning har biri ushbu faoliyat uchun mas'ul bo'lgan muassasaning tarkibiy bo'linmasini ifodalaydi.

- **Shtatlar (shtat mashinasi).** Ushbu turdagi diagrammaning yana bir nomi Davlat mashinasi diagrammasi. Ushbu turdagi diagrammalar, aslida, dasturiy ta'minot tizimining xatti-harakatlari ko'rsatilgan cheklangan holat mashinasidir. Davlat mashinasidan farqli o'laroq, holat diagrammasi kompozit deb ataladigan holatlarga ham ega bo'lishi mumkin. Avtomat diagrammasining holati holatni bir-biridan ajratish uchun ishlatiladigan nomga ega bo'lishi mumkin, bu holatga kirish (chiqish) paytida bajariladigan harakatlar, ichki o'tishlar va pastki holatlar, agar holat kompozit bo'lsa, kechiktirilgan hodisalar ro'yxati. bu holatda qayta ishlanmagan va boshqa shtatlarda qayta ishlash uchun navbatda turgan voqealar. Holat diagrammasi kontekstida hodisa bir holatdan ikkinchi holatga o'tishni boshlaydigan stimuldir. Hodisa, masalan, ob'ekt kutilayotgan holatdan yangi holatga o'tgan vaqt oralig'ining tugashi bo'lishi mumkin. Hodisaning yana bir misoli - foydalanuvchi tugmani bosganda yoki tarmoq orqali ma'lumotlar paketi kelganda kompyuterda apparat uzilishi.

- **Foydalanish holatlari.** Bu aktyorlar va foydalanish holatlari o'rtasidagi munosabatlarni tavsiflaydigan diagrammalar

- **Foydalanish sxemasining** asosiy maqsadi mijoz, oxirgi foydalanuvchi va ishlab chiquvchiga dasturiy ta'minot tizimining funkcionalligi va xatti-harakatlarini birgalikda muhokama qilish uchun vositani taqdim etishdir. Foydalanish diagrammasi odatda quyidagilarni o'z ichiga oladi:

- *Pretsedentlar.* Foydalanish holati dasturiy ta'minot tizimidan tashqi foydalanuvchi tomonidan qanday foydalanishni anglatadi. Misol uchun, agar dasturiy ta'minot tizimi uyali telefon bo'lsa, unda quyidagi pretsedentlar bo'lishi mumkin: qo'ng'iroq qilish, qo'ng'iroq qilish, elektron pochmani o'qish, qo'shimcha qo'ng'iroqni qabul qilish va rejalashtiruvchidan foydalanish.

- *Aktyorlar.* Bu dasturiy ta'minot tizimining tashqi foydalanuvchilari sinflari. Sinflar rollar bo'yicha tashkil etiladi, ularda odatda kamroq rollar mavjud. Masalan, uyali telefon uchun faqat ikkita aktyor mavjud: foydalanuvchi va uyali tarmoq.

- *bog'liqlik, umumlashtirish va assotsiatsiya munosabatlari.* Foydalanish holatlari va aktyorlar o'rtasida turli munosabatlar kiritilishi mumkin. Misol uchun, "qo'shimcha qo'ng'iroqni qabul qilish" pretsedenti "qo'ng'iroqni qabul qilish" foydalanish holatiga bog'liq, chunki qo'shimcha qo'ng'iroq faqat qo'ng'iroq allaqachon foydalanuvchi tomonidan qabul qilingan bo'lsa qabul qilinishi mumkin. Bank dasturiy ta'minot tizimi uchun siz mijozlar turlarini kiritishingiz mumkin: individual mijoz va korporativ mijoz. Bu, shubhasiz, bir tomondan, "individual mijoz" va "korporativ mijoz" aktyorlari va ikkinchi tomondan, aktyor "mijoz" o'rtasida umumlashtiruvchi munosabatni joriy etishni talab qiladi.

Yuqoridagi diagrammalarning tavsifidan ko'rinib turibdiki, tizimning xatti-harakati UMLda uchta usulda tavsiflanadi: tizimning faoliyati ko'rib chiqiladi, u muassasaning tarkibiy bo'linmalari ishining tavsifini va tashkilotni o'z ichiga oladi. dasturiy ta'minot tizimining bir nechta komponentlarining parallel ishlashi, ushbu sinf ob'ektlarining xatti-harakatlari ko'rsatilgan, ular orasidagi holatlar va o'tishlar shaklida tashkil etilgan va tashqi ta'sirlar bilan boshlangan butun dasturiy ta'minot tizimining xatti-harakati ham ko'rib chiqiladi.

O'zaro ta'sir diagrammalari

UMLda o'zaro ta'sir diagrammalarining asosiy turlari ketma-ketlik va hamkorlik diagrammalaridir. O'zaro ta'sir diagrammasi ob'ektlar to'plamini va ular o'rtasidagi munosabatlarni, shu jumladan ob'ektlar o'rtasida almashinadigan xabarlarni o'z ichiga olgan munosabatlarni belgilaydi. Ketma-ketlik diagrammalari xabarlarning vaqtinchalik tartiblanishiga, hamkorlik diagrammalari esa xabarlarni jo'natish va qabul qilish ob'ektlarining strukturaviy tashkil etilishiga qaratilgan. O'zaro ta'sir diagrammalari dinamik jarayonlarni modellashtiradi va bu o'zaro ta'sir diagrammalarining kichik turlariga mos keladigan ikki xil usulda amalga oshirilishi mumkin. Yaxshiroq tushunish uchun misolni ko'rib chiqing. Jarayon kino bo'lsin. Bir tomondan, kino plyonkadagi ramkalar ketma-ketligi, ya'ni odatda ketma-ketlik diagrammasi bilan ifodalangan narsa. Boshqa tomondan, filmni suratga olish guruhi yaratdi, ssenariy yozildi, sahnalar, manzaralar va hokazolar rejalashtirilgan. narsalar hamkorlik sxemasiga o'xshash. Ketma-ketlik diagrammalarida o'zaro ta'sirda

ishtirok etuvchi ob'ektlar X o'qi bo'ylab joylashtirilgan, o'zaro ta'sirni boshlagan ob'ekt chap tomonga, qolgan ob'ektlar o'ngga joylashtirilgan. Ob'ektlar tomonidan yuborilgan va qabul qilingan xabarlar keyin y o'qi bo'ylab joylashtiriladi, bu xabarlar vaqt tartibida - keyingi xabarlar quyida paydo bo'ladi. Har bir ob'ekt uchun ketma-ketlik diagrammasi uning hayot chizig'ini ko'rsatadi. Bu ob'ektning vaqt ichida mavjudligini aks ettiruvchi vertikal chiziqli chiziq. Ko'pgina ob'ektlar o'zaro ta'sir davomida mavjud, shuning uchun ularning hayot chiziqlari diagrammaning pastki qismiga chizilgan. Hayot chiziqlarining boshlanishi odatda boshqacha bo'ladi, chunki ob'ektlarni yaratish ko'pincha boshqa ob'ektlar tomonidan boshlanadi. Shuningdek, nazorat markazi deb ataladigan narsa ob'ektlarning hayot chizig'iga tortiladi. U cho'zilgan to'rtburchak sifatida tavsiflangan bo'lib, u ob'ekt harakatni bajaradigan vaqt davrini ko'rsatadi. To'rtburchakning yuqori qirrasini vaqt o'qi bo'ylab harakat boshlangan paytga, pastki qismi esa tugash vaqtiga to'g'ri keladi. Hamkorlik diagrammasi o'zaro ta'sirda ishtirok etuvchi ob'ektlarni tashkil etishga qaratilgan. Ushbu turdagi diagrammalardagi ob'ektlar grafik uchlari ko'rinishida joylashgan va ob'ektlar orasidagi bog'lanishlar ushbu grafikning yo'ylari bilan tavsiflangan. Havolalar, shuningdek, ob'ektlar qabul qiladigan va yuboradigan xabarlar bilan to'ldiriladi. Hamkorlik diagrammalari ularni ketma-ketlik diagrammalaridan ajratib turadigan ikkita xususiyatga ega:

1. **Yo'l.** Bir ob'ektning boshqasiga bo'lgan munosabatini tavsiflash uchun ushbu munosabatlarning eng uzoq nuqtasiga yo'l stereotipi (masalan, teglangan ob'ekt xabar jo'natuvchiga mahalliy ekanligini ko'rsatadigan mahalliy) biriktirilishi mumkin.

2. **Xabarning seriya raqami.** Xabar oldidan vaqt ketma-ketligini ko'rsatish uchun siz har bir yangi xabar uchun ko'payishi kerak bo'lgan raqamni qo'yishingiz mumkin.

Ketma-ketlik diagrammalari va hamkorlik diagrammalari bir xil ma'lumotni faqat bir oz boshqacha burchaklardan tavsiflaydi. Demak, bu turdagi diagrammalar semantik jihatdan ekvivalent deb aytishimiz mumkin. Bu shuni anglatadiki, bir turdagi diagramma ma'lumotni yo'qotmasdan boshqa diagrammaga aylantirilishi mumkin. Bu, albatta, har ikkala turdagi diagrammalarda bir xil ma'lumotlar taqdim etilganligini anglatmaydi. Ketma-ketlik diagrammasi va hamkorlik

diagrammasi bir xil modeldan foydalanadi, lekin uning turli xususiyatlarini ingl. UML 2.0 shuningdek, o'zaro ta'sir diagrammalarining yangi turlarini taqdim etadi:

- O'zaro ta'sirni ko'rib chiqish diagrammasi - bu ketma-ketlik diagrammasi fragmentlari va boshqaruv oqimi konstruksiyalarini o'z ichiga olgan faoliyat diagrammasining bir turi.

- Vaqt - bu ma'lum vaqt shkalasi bilan hayot chizig'i bo'ylab holatdagi o'zgarishlarni aniq ko'rsatadigan ketma-ketlik diagrammasining muqobil ko'rinishi. Haqiqiy vaqtda ilovalarda foydali bo'lishi mumkin.

Albatta, bu yerda tavsiflangan UML tavsifidan uni qo'llash metodologiyasini to'liq tushunish qiyin, ammo bu mavzu bo'yicha juda ko'p monografiyalar mavjud. Katta loyihalar uchun ularning tavsiflarini shakllantirish dizaynning eng ko'p vaqt talab qiladigan qismidir (dasturlash bilan solishtirganda). Bundan tashqari, UML tilidagi tavsifga ko'ra, ushbu tavsiflarga mos keladigan dastur matnlari avtomatik ravishda yaratiladigan bir nechta tizimlar mavjud. Biroq, dasturlarni tuzatish bosqichida dasturchilar loyihaning UML hujjatlarida avtomatik ravishda amalga oshirilmaydigan dasturlarga o'zgartirishlar kiritishlari mumkin. Shuning uchun hujjatlarni sinxronlashtirish uchun dizayn intizomini saqlash talab qilinadi.

Gruber metodologiyasi

Ushbu bo'limda biz Tomas Gruber tomonidan "Bilim almashinuvi uchun ishlatiladigan ontologiyalarni loyihalashning taxminiy tamoyillari" maqolasida taklif qilingan ontologiyani qurish metodologiyasini ko'rib chiqamiz. O'quvchilarning ko'pchiligi, ehtimol, relyatsion ma'lumotlar bazalarini yaratish tamoyillari bilan tanish bo'lishi mumkin: ma'lumotlar bazasi asta-sekin normal shaklga keltiriladi va har bir bosqichda ma'lum bir transformatsiya amalga oshiriladi - ma'lumotlar bazasi oddiy shakllar deb ataladigan shaklga aylantiriladi. Xuddi shunday metodologiya Gruber tomonidan ontologiyalarni qurish uchun taklif qilingan bo'lib, u ontologiyalarni quyidagi printsiplarga asoslanib loyihalashni taklif qilgan:

1. **Aniqlik.** Ontologiya, iloji boricha, u rasmiylashtirish bo'lgan kontseptual sxemaning mazmunini aks ettirishi kerak. Iloji bo'lsa, qisman emas, balki to'liq ta'riflardan (ontologiya elementlari orasidagi munosabatlar) foydalanish afzalroqdir. Barcha ta'riflarni tabiiy tilda hujjatlashtirish tavsiya etiladi.

2. **Muvofiqlik** (Muvofiqlik). Ontologiya mos bo'lishi kerak, ya'ni tushunchalarning ta'rifi va ular o'rtasidagi munosabatlardan kelib chiqadigan xulosalar dastlabki ta'riflarga mos kelishi kerak. Norasmiy tavsiflangan tushunchalar uchun ham moslik saqlanishi kerak. Agar rasmiy ta'riflardan olingan xulosalar norasmiy tavsiflarga mos kelmasa, ontologiya mos kelmaydigan deb hisoblanadi.

3. **Uzayuvchanlik**. Ontologiya umumiy ontologiya kutubxonalarida qo'shimcha harakatlarsiz ishlatilishi mumkin bo'lgan tarzda ishlab chiqilishi kerak. Bunday dizaynning eng muhim shartlaridan biri bu ontologiyadagi mavjud elementlarga asoslanib, ikkinchisini o'zgartirishni talab qilmaslik uchun yangi tushunchalarni belgilash qobiliyatidir.

4. **Minimal kodlash tarafkashligi**. Loyihalashtirilgan kontseptual sxema rasmiylashtirilgan tavsifni yozish uchun ishlatiladigan o'ziga xos tilga bog'liq bo'lmasligi kerak. Kodlashga bog'liqlik, ontologik tavsifni tanlash ontologiya yozilgan tilning yozuvlari bilan mos kelishiga asoslangan bo'lsa paydo bo'ladi. Boshqa tillardan foydalanadigan turli ontologiya asoslari prognoz qilingan ontologiyani osongina tushunishi uchun bu bog'liqlikni minimallashtirish kerak.

5. **Minimal ontologik majburiyat**. Ontologiya imkon qadar kamroq modellashtirilayotgan dunyo ontologiyasi haqidagi faktlarni o'z ichiga olishi kerak, bu esa bu ontologiyani boshqa ontologiyalarda qo'llash erkinligini ta'minlashi kerak. Agar muammoning kontseptual sxemasi dunyo ontologiyasining tavsifi muhim ahamiyatga ega bo'lsa, unda bu tavsif imkon qadar minimal bo'lishi kerak. U faqat tushunchalar atamalarining ro'yxati bilan cheklanishi kerak, ular orasidagi munosabatlarni aniqlamasdan, ya'ni Gruber nuqtai nazaridan "zaif" nazariyani tuzing. Shunda dunyo ontologiyalarini o'ziga xos tarzda belgilaydigan turli ontologiya asoslari bu tushunchalarga o'ziga xos ma'no bera oladi.

Keling, illyustrativ misolni ko'rib chiqaylik. Muhandislik tizimlarini modellashtirish amaliyotda keng tarqalgan vazifadir. Muhandislar fizik tizimlarning xatti-harakatlarini ifodalash uchun tenglamalar tizimi kabi turli matematik modellardan foydalanadilar. Fizik

tizimlarning kontseptual sxemalarini va ularning matematik modellarini tavsiflovchi ontologiyadagi eng muhim tushunchalar:

- Jismoniy miqdor: 3 metr, soatiga 80 kilometr.
- Jismoniy o'lchov: uzunlik yoki uzunlik / vaqt.
- O'lchov birligi: metr, soatiga kilometr.
- Turli tartibli kattaliklar: skalyarlar, vektorlar, tenzorlar va ularning funksiyalari.
- Turli matematik cheklovlarni tavsiflash uchun algebralar (algebralar): ko'paytirish, yig'ish, darajaga ko'tarish va boshqalar.
- Matematik ifodalarni muammo sohasi ob'ektlari sifatida tavsiflash uchun metalingvistik darajadagi tushunchalar. Masalan, tenglamalar tizimida bog'liq o'zgaruvchilarni tavsiflashingiz mumkin.

Keling, fizik miqdor tushunchasini tavsiflashga harakat qilaylik. Ushbu misolni tavsiflash uchun biz bu yerda metalingvistik tavsiflardan foydalanmaymiz. Biz bunday ontologiyaning bir nechta variantlarini keltiramiz, eng oddiyidan boshlab va ontologiyaning xususiyatlarini variantdan variantga takomillashtirish va yuqoridagi dizayn tamoyillari asosida. Ontologiyalarni qurish uchun biz KIF tilidan foydalanamiz, uning qisqacha tavsifi yuqorida keltirilgan.

Variant 1. Jismoniy miqdor juftlik (raqam, o'lchov birligi) bilan ifodalanadi. Raqam ikki tomonlama aniqlikdagi suzuvchi nuqta formatida ifodalanadi, ya'ni qo'sh float tipidagi bo'ladi. Biz fizik miqdorni qiymat va o'lchov birligidan iborat ob'ekt sifatida aniqlaymiz, buning uchun ikkita bir o'rinli funktsiyani kiritamiz: miqdor-kattalik va miqdor-birlik. Fizik-miqdor tipining ta'rifi, shuningdek, aniqlangan predikat bilan ko'rsatilgan fizik miqdorning qiymati va birligi mavjudligini talab qiladi. Shuningdek, fizik miqdor "o'shanda va faqat keyin" (\Leftrightarrow) sharti orqali ko'rsatilganligi sababli, biz har bir bunday qiymatlar juftligi va o'lchov birliklarini fizik miqdor deb hisoblaymiz. Biz ta'rifda o'lchov birliklarini aniq sanab o'tamiz.

(defrelation physical-quantity

(<=> (physical-quantity ?q)

(and (defined (quantity.magnitude ?q))

(double-float (quantity.magnitude ?q))

(defined (quantity.unit ?q))

(member (quantity.unit ?q))

(metr to'plami ikkinchi kilogramm amper kelvin mol kandela))))

Demak, "fizik-miqdor" - bu sinf (sinfning barcha holatlari uchun bir o'rinli munosabatlar). Ushbu sinf misollarini tavsiflash uchun siz konstruktorni "miqdori" deb nomlangan funktsiya sifatida belgilashingiz kerak. Ifoda ("? M" "? U" miqdori) "? M" qiymati va o'lchov birligi "? U" bilan "? Q" fizik miqdorini anglatadi.

(deffunction the-quantity

(<=> (physical-quantity ?q)

(and (defined (the-quantity ?m ?u))

(= (the-quantity ?m ?u) ?q)

(defined (quantity.unit ?q))

(and (physical-quantity ?q)

(= (quantity.magnitude ?q) ?m)

(= (quantity.unit ?q) ?u))))

Jismoniy miqdorning ma'lum bir misoli ko'rsatilishi mumkin, masalan:

(= X (fizik miqdori 3 metr))

Bir qarashda fizik miqdor tushunchasining yuqoridagi ta'rifi to'g'ri ko'rinadi. Ammo ma'lum bo'lishicha, bu ta'rif yuqorida keltirilgan ontologiyani loyihalash tamoyillariga mos kelmasligi sababli sezilarli kamchiliklarga ega.

1. Ta'rifda o'rnatilgan juft float turining qiymatlari qo'llaniladi. Ushbu turdagi ontologiya qaysi turdagi mashinalarda amalga oshirilishiga yoki uni amalga oshirishda dasturchilar tomonidan aniqlangan aniqlik miqdoriga qarab turli xil usullarda turli xil ontologiyalarda kodlanishi mumkin. Bu, shubhasiz, kodlashga minimal bog'liqlik tamoyiliga ziddir.

2. O'lchov birligi tushunchasi yuqorida miqdor.birlik funktsiyasi uchun mumkin bo'lgan qiymatlar to'plami sifatida belgilangan. Bu, shuningdek, kodlashga minimal bog'liqlik tamoyilini buzadi, lekin avvalgi holatdan ko'ra nozikroqdir. Demak, o'lchov birligi tushunchasining mumkin bo'lgan ma'nolari to'plamining ta'rifi fizik miqdor tushunchasining ta'rifida kodlangan.

3. Shuningdek, o'lchov birligining mumkin bo'lgan qiymatlari to'plamining qat'iy ta'rifi ontologiyaning kengayishini cheklaydi. O'lchov birligi tushunchasini aniqlash kerak, shunda yangi o'lchov birliklarini qo'shish oson bo'ladi.

Variant 2. Yuqoridagi kamchiliklarni qanday bartaraf etishni ko'rib chiqamiz. Birinchidan, qiymat va birlik tushunchalari mustaqil ravishda aniqlanishi uchun fizik-miqdorning ta'riflarini o'zgartiramiz. Buning uchun biz ikkita mustaqil sinfni kiritamiz: *magnitude* va *unit-of-measure*.

(defrelation physical-quantity
(<=> (physical-quantity ?q)
(and (defined (quantity.magnitude ?q))
(magnitude (quantity.magnitude ?q))
(defined (quantity.unit ?q))
(unit-of-measure (quantity.unit ?q))))))

Haqiqiy sonlar sinfini kiritish uchun *magnitude* sinfini aniqlaymiz. Haqiqiy raqamlar Ontolingua ontologiya kutubxonasida aniqlangan haqiqiy sonlar sinfi bilan ifodalanadi. Ushbu ta'rif *double float* turini kodlashga bog'liqlikdan saqlaydi:

(defrelation magnitude
(<= (magnitude ?x)
(real-number ?x)))

E'tibor bering, kattalik sinfining ta'rifi to'liq emas - zarur (<=), lekin yetarli emas. Unda aytilishicha, barcha haqiqiy raqamlar qiymatdir, lekin u barcha qiymatlar haqiqiy sonlar deb aytilmaydi. Shuning uchun sinf boshqa turdagi qadriyatlarni kiritish uchun ochiqdir. O'lchov birligining ta'rifi biroz murakkabroq. Birinchidan, biz ibtidoiy (aksioma ta'rifi yo'q) sinfni aniqlaymiz *unit-of-measure*:

(defrelation unit-of-measure
(class unit-of-measure))

Endi, ushbu sinfga asoslanib, siz o'lchov birliklarining qiymatlari to'plamini kengaytirish mexanizmini yaratishingiz mumkin. Buning uchun biz asosiy birlik sinfini kiritamiz, uning misollari birlik qiymatlari bo'ladi. Bunday qiymatlar uchun biz bir vaqtning o'zida ikkita operatsiyani aniqlaymiz: mavjud bo'lganlardan yangi o'lchov birliklarini qurish birligi * (ko'paytirish) va ^ birligi (eksponentsiya). Bu, masalan, tekis raqamlar maydonini (kvadrat metr) yoki tezlikni (sekundiga metr) o'lchash kerak bo'lganda foydalidir, buning uchun eksponentatsiya operatsiyasi salbiy ko'rsatkichlarni (bo'linish) ko'rsatish orqali o'lchov birliklarini aniqlashga imkon beradi.

(defrelation basic-unit
(=> (basic-unit ?u) ; asosiy basic-unit boshqacha

```

(unit-of-measure ?u))) ; unit-of-measure misollaridan
(deffunction unit*
; unit* unit-of-measure har bir juftligiga
; unit-of-measure ga mos keladi
(=> (and (unit-of-measure ?u1)
(unit-of-measure ?u2))
(and (defined (unit* ?u1 ?u2))
(unit-of-measure (unit* ?u1 ?u2))))
; operatsiya kommutativdir
(= (unit* ?u1 ?u2) (unit* ?u2 ?u1))
; operatsiya assotsiativdir
(= (unit* ?u1 (unit* ?u2 ?u3))
(unit* (unit* ?u1 ?u2) ?u3))
(deffunction unit^
; unit^ har bir juftlik unit u real
; unit-of-measure mos keladi
(=> (and (unit-of-measure ?u)
(real-number ?r))
(and (defined (unit^ ?u ?r))
(unit-of-measure (unit^ ?u ?r))))
; algebraik xossalari qanoatlantiriladi
; eksponentatsiya operatsiyalari
(= (unit^ ?u 1) ?u)
(= (unit* (unit^ ?u ?r1) (unit^ ?u ?r2))
(unit^ ?u (+ ?r1 ?r2)))
(= (unit^ (unit* ?u1 ?u2) ?r)
(unit* (unit^ ?u1 ?r) (unit^ ?u2 ?r)))

```

Endi siz **basic-unit** ni asosiy birlik sinfining misollari sifatida belgilashingiz mumkin:

```

(defobject metr
(basic-unit metr))
(defobject sekund
(basic-unit sekund))
(defobject metr/sekund
(= metr/sekund
(unit* metr (unit^ sekund -1))))

```

Yuqoridagi ta'rif etarlicha yaxshi ko'rinadi, qiymatlar va birliklar osongina kengaytiriladi, buning uchun siz asosiy birlik sinfining yangi

namunasini belgilashingiz yoki kattalik sinfi qiymatlariga yangi tur qo'shishingiz kerak. Xato topilishi mumkin bo'lgan yagona narsa - bu turli o'lchov birliklarining mos kelmasligi. Bizning ta'rifimizda, masalan, 6 santimetr 60 millimetrga teng, ya'ni hech qanday tarzda ko'rsatilmagan. ular ontologiyada bir xil mavjudotlar emas. Buni ontologiyaning ta'rifini mos ravishda o'zgartirish orqali ham tuzatish mumkin. Muallif o'quvchini ushbu o'zgartirishni mashq sifatida mustaqil ravishda amalga oshirishga taklif qiladi. Biz shuni ta'kidlaymizki, to'liq tavsif uchun ontologiyaga jismoniy o'lchov tushunchasini kiritish kerak bo'ladi.

BORO usuli

BORO (*Business Object Reference Ontology*) - o'tgan asrning 90-yillari o'rtalarida Kris Partridge tomonidan mavjud axborot tizimlarini axborot modellariga qayta tashkil etish muammosini hal qilish uchun ishlab chiqilgan ontologiyalarni qurish metodologiyasi. Kris Partidj bir necha yil davomida ushbu muammo ustida ishlamoqda va natijada eskirgan tizimli tashkilotga ega bo'lgan bunday axborot tizimlari uchun engil ontologiyalarni (tasniflarni) qurish metodologiyasini ishlab chiqdi. Ushbu uslub BORO metodologiyasi deb nomlandi va soddaligi va oddiyligi tufayli mashhurlikka erishdi. Ushbu texnikaga asoslanib, yuqorida muhokama qilingan "*IDEAS*" kabi taniqli ontologiya kutubxonasi ishlab chiqilgan. Mavjud dasturiy ta'minot tizimlarini qayta tashkil etish vazifasi juda keng tarqalgan. Uskunalar tezda eskiradi, tizim o'sib boradi va mavjud arxitektura yangi muammolarga javob berishni to'xtatadi. Shunday qilib, dasturiy ta'minot tizimini qayta tashkil etish muammosi paydo bo'ladi.

Mavjud axborot tizimini qayta tashkil etish vazifasini taxminan ikki bosqichga bo'lish mumkin:

1. **Teskari muhandislik.** Ushbu bosqichda mavjud tizim tahlil qilinadi. Biznes ob'ektlari kod va ma'lumotlardan olinadi va hujjatlantiriladi. Shunday qilib, ushbu bosqichda amalga oshirilayotgan ishlab chiqarish jarayoni ushbu tizimni ishlab chiqish jarayonida amalga oshirilganidan teskari yo'nalishda ketadi. Shuning uchun bu bosqich teskari muhandislik deb ataladi.

2. **Forvard muhandisligi.** Ushbu bosqichda oldingi bosqichda qurilgan mavjud tizimning biznes modeli asosida eskisini almashtirish uchun yangi axborot tizimining yanada nozik arxitekturasi ishlab chiqilgan. Bu jarayon allaqachon umumiy muhandislik faoliyati bo'lib,

u berilgan vazifaning kontseptual sxemasini bilishdan iborat. Shuning uchun bu bosqich bevosita muhandislik deb ataladi.

Ikkala bosqich ham birgalikda reinjiniring deb ataladi.

BORO usuli - bu reinjiniring jarayonini soddalashtirish uchun tamoyillar va qoidalar to'plami. Bu yerda odatiy muhandislik modellashtirish jarayonidan asosiy farqi shundaki, modellashtirish qayta tashkil etilgan dasturiy tizimning mavjud kontseptual sxemasi asosida amalga oshiriladi. **BORO** usuli modellashtirilgan biznes-jarayonning kontseptual diagrammasi va dasturiy ta'minot tizimining mantiqiy asosi bo'lgan ishlab chiqarish jarayonining kontseptual diagrammasi o'rtasida farqlanadi. **BORO**dagi oxirgi metodologiya biznes jarayonining kontseptual diagrammasidan ko'ra to'liqroq va chuqurroq hisoblanadi. Shu ma'noda, **BORO** metodologiyasi ob'ektga yo'naltirilgan yondashuvni tanqid qiladi, bu biznes jarayonining kontseptual diagrammasi va uning dasturlash modeli o'rtasida to'g'ridan-to'g'ri muvofiqlikni beradi. Ushbu yondashuv bilan kontseptual model tushunchalarini osongina noto'g'ri aniqlash mumkin. **BORO** metodologiyasi berilgan topshiriqning kontseptual asoslari tushunchalarini qanday aniqlash mumkinligi bo'yicha ko'rsatmalar to'plamini taqdim etadi. **BORO** metodologiyasi tushunchalarni Aristotel an'alariga ko'ra ko'rib chiqadi, ya'ni atribut xususiyatlari to'plami sifatida. Bir xil atributlarga ega bo'lgan ob'ektlar bir xil tushuncha-sinfga tegishli deb hisoblanadi. **BORO** metodologiyasi sinflarni to'g'ri aniqlash muammosini hal qiladi.

Buning uchun siz quyidagilarni bajarishingiz kerak:

1. Mavzu sohasi sub'ektlarini (jismoniy shaxslarini) tanlang. Tanlov asosiy mezon, fazoviy-vaqtinchalik ko'lami asosida amalga oshiriladi. Har qanday individ fazoviy yoki vaqtinchalik darajada ifodalangan substansiya (Aristotel ma'nosida) asosida mavjud bo'ladi, deb ishoniladi.

2. Sinflarni tanlang. Kontseptual sxemaning fazoviy va vaqtinchalik kengayishiga ega bo'lmagan har qanday ob'ekt alohida shaxs emas deb hisoblanadi. Lekin har bir bunday ob'ekt sinf emas.

3. Mavzu sohasi elementlari orasidagi munosabatni ajratib ko'rsating. Aloqalar fazoviy-vaqtinchalik kengayishiga ega bo'lmagan va misollarga ega bo'lmagan ob'ektlardir.

Yuqoridagi printsiplarga asoslanib, u quyidagi algoritmni taklif qiladi:

Algoritm 2.1. BORO ontologiyasini qurish usuli

Kirish: Axborot tizimi, uning asosida axborot modelini qurish kerak.

Natija: Ontologiya shaklida yozilgan axborot modeli, uch turdagi ob'ektlarni o'z ichiga oladi: shaxslar, sinflar va munosabatlar (kortejlar).

1-qadam. Mavzu sohasi elementini tanlang. Agar hali ham ko'rib chiqilmagan elementlar mavjud bo'lsa, unda 2-bosqichga o'ting. Aks holda, ishni tugating.

Qadam 2. Agar element fazoviy va vaqtinchalik kengayishga ega bo'lsa, unda biz uni ontologiyaga individual ravishda qo'shamiz va 1-bosqichga o'tamiz. Aks holda, 3-bosqichga o'ting.

3-qadam. Agar elementda misollar bo'lsa, biz uni sinf sifatida ontologiyaga qo'shamiz, uning misollarini aniqlaymiz va 1-bosqichga o'tamiz. Aks holda, 4-bosqichga o'tamiz.

4-qadam. Agar elementda misollar bo'lmasa, bu munosabatlardir. Biz uni kortej sifatida yozamiz, ushbu kortejdagi ob'ektlarni aniqlaymiz va uni ontologiyaga qo'shamiz. Keyin 1-bosqichga o'ting.

Yuqoridagi algoritmdan ko'rinib turibdiki, qayta tashkil etilgan axborot tizimining kontseptual modelining ontologiyasi elementlarini aniqlash jarayoni iterativdir. Olingan ontologiya axborot tizimining kontseptual sxemasi elementlarining fazoviy-vaqt xususiyatlarini hisobga olgan holda dunyo ontologiyasiga kiritiladi. Shunday qilib, tuzilgan ontologiyada Gruber ontologiyalarini qurishning beshinchi printsipli, minimal ontologik fiksatsiya bajarilmaydi. Tanlash jarayoni qiyin emas, lekin natijada paydo bo'lgan ontologiya odatda oddiy tasnif bo'lib, har doim hech qanday munosabatlarni o'z ichiga olmaydi. Garchi umumiy holatda BORO usuli modellashtirish uchun qo'llanilmasa ham, biznes loyihalarini qurish tasnifining alohida holatlari uchun uni qo'llash juda mumkin. BORO metodologiyasini to'liqroq tushunish uchun tavsiflovchi misolni ko'rib chiqing. Unda jadval ko'rinishidagi mamlakatlarning bir-biri bilan chegaralari haqidagi ma'lumotlarni o'z ichiga olgan axborot tizimi bo'lsin, uning kichik qismi 2.3-rasmda ko'rsatilgan. Keling, uning asosida BORO metodologiyasiga ko'ra ontologiyani quraylik. Jadvalning birinchi ustunida Rossiya, Argentina va Belorussiya tipidagi ob'ektlar mavjud bo'lib, ular fazo-vaqt darajasiga ega. Shuning uchun biz ularni individual deb hisoblaymiz. Bu shaxslar "mamlakat" sinfining

misollari sifatida ko'rib chiqiladi - birinchi ustunning birinchi qatorida joylashgan ob'ekt. Bu ob'ektning kengaytmasi yo'q, lekin uning misollari bor va shuning uchun sinf hisoblanadi. Ikkinchi ustunda jismoniy shaxslar hisoblangan 5529 km, 9861 km va 2900 km turdagi ob'ektlar mavjud. Bular "chegara" sinfining misollari. Nihoyat, uchinchi ustunda ikki davlat chegaralarini bog'laydigan juftliklar ro'yxati mavjud.

Shunday qilib, axborot tizimi bizga quyidagi shakldagi faktlarni beradi:

- *Davlatlar bor.*
- *Ularning har birining chegarasi bor.*
- *Mamlakatlar chegaralarining ba'zi qismlari bir-biriga to'g'ri keladi.*

Biz allaqachon "mamlakat" va "chegara" sinflarini e'tiborga olganmiz. Ayrim mamlakatlar chegaralari qismlarining mos kelishini qanday aniqlash mumkin? Buning uchun biz "chegara" sinfining ikkita kichik turini kiritamiz: "mamlakat chegarasi" va "umumiy chegara". Mamlakat turi mamlakat chegarasi turi bilan mamlakat sinfining maxsus atributi bilan ifodalanishi mumkin bo'lgan qisman butun munosabat bilan bog'lanadi. "Mamlakat chegarasi" turi va "umumiy chegara" turi ham bir qism-butun munosabat bilan bog'langan. Shunday qilib, natijada to'rtta sinf mavjud: "mamlakat", "chegara", "mamlakat chegarasi" va "umumiy chegara". "Chegara" sinflari va "mamlakat chegarasi" va "umumiy chegara" sinflari bir-biri bilan taksonomiya munosabati bilan, "mamlakat" sinfi "mamlakat chegarasi" sinfiga qisman-butun munosabat bilan, shuningdek sinfi sifatida "mamlakat chegarasi" sinfi bilan "umumiy chegara". Misol uchun, agar mamlakat sinfining ikkita misoli mavjud bo'lsa: Fransiya va Belgiya, u holda mamlakat chegarasi sinfining tegishli misollari mavjud. Fransiya va Belgiya umumiy chegaraga ega bo'lganligi sababli, "davlat chegarasi" sinfining tegishli holatlari bilan qisman munosabatlar bilan bog'liq bo'lgan "umumiy chegara" sinfining misoli ham bo'ladi, bu munosabatlar mamlakatlarni belgilaydi. umumiy chegaraga ega.

2.2. OWL – WEB ontologiyalari uchun tavsif qilish tili

OWLga bo'lgan ehtiyoj

Ushbu bo'limda biz Internet (World Wide Web) ontologiyasini tavsiflash uchun yaratilgan OWL (Web Ontology Language) tilini ko'rib chiqamiz. World Wide Web - bu Internet orqali kirish mumkin bo'lgan giperbog'langan hujjatlar tizimi. Gipermatnli hujjat oddiy hujjatdan farq qiladi, chunki u nafaqat matnni, balki boshqa ma'lumotlarni ham o'z ichiga olishi mumkin: rasmlar, musiqa va video fayllar va boshqalar. Bunday hujjatlarni maxsus dastur - veb-brauzer yordamida ko'rish mumkin. Muallifning fikricha, ushbu monografiyaning har bir o'quvchisi hech bo'lmaganda bir marta veb-brauzer orqali gipermatnli hujjatlarni ko'rgan, shuning uchun Internet mafkurasining batafsil taqdimoti bu yerda nomaqbul vazifaga o'xshaydi. Keling, World Wide Web vorisi - aqlli Internet yoki Semantik Internetni ko'rib chiqishga yaxshiroq e'tibor qarataylik. Semantik Internet butunjahon tarmog'ining keyingi avlodi bo'lib, u gipermatnga qo'shimcha ravishda ushbu hujjatlarning semantikasining tavsiflarini, shuningdek, ushbu hujjatlarni oxirgi foydalanuvchilarga taqdim etuvchi turli xizmatlarning semantikasining tavsiflarini o'z ichiga oladi. Bunday tavsiflar nafaqat odamlar uchun, balki kompyuterlar uchun ham tushunarli bo'lishi kerak, aks holda bunday tavsiflarning ma'nosi kam bo'ladi. Kompyuterlar tomonidan veb-hujjatlarning mazmunini tushunish dasturlarga Internetda ma'lumotni aniqroq qidirish, bu ma'lumotlarni avtomatik ravishda tartibga solish va boshqa dasturlar bilan bunday ma'lumotlarni almashish imkonini beradi. Odatda, Semantik Internet Internetning yaqinlashib kelayotgan versiyasi - Web 3.0 deb ataladigan komponent sifatida tavsiflanadi. Hozircha biz faqat Web 3.0 qanday bo'lishini taxmin qilishimiz mumkin, ammo uning asosiy komponentlaridan biri Web-hujjatlari semantikasining tavsifini nazarda tutuvchi Semantik Web bo'lishi aniq. W3 Konsortsiumi Semantik Internetning butun mafkurasini ishlab chiqdi, shu jumladan veb-hujjatlarning semantikasini tavsiflash uchun ishlatiladigan tillarning ta'rifi, ularni standartlashtirish, shuningdek, muayyan hujjatlarni tavsiflash uchun ushbu tillardan foydalanish bo'yicha takliflar. Konsortsiyum veb-saytida Semantik Internetga bag'ishlangan maxsus bo'lim mavjud. Veb-axborot semantikasini tavsiflash uchun RDF (Resurs tavsifi ramkasi) tilidan

foydalaniladi, biz uni quyida batafsil ko'rib chiqamiz. Bu til veb-hujjatlarning resurslarini tavsiflash uchun mo'ljallangan, ammo bu to'liq ontologiyalarni ifodalash uchun yetarli emas. Shu sababli, Web ontologiyasini tavsiflash uchun maxsus til, OWL tili ham taklif qilindi. Ushbu til RDF tiliga mos keladi, ya'ni har bir OWL hujjati ham RDF hujjatidir. RDF XML sintaksisi asosida qurilgan bo'lib, u Internetda ma'lumotlar almashinuvi uchun standart sifatida ishlatiladi. Shuning uchun har bir OWL hujjati ham haqiqiy XML hujjatidir. Keyinchalik, biz XML tamoyillarini ham ko'rib chiqamiz. O'quvchi, ehtimol, amaliy ontologiyalarga bag'ishlangan monografiyada OWL tilining tavsifi nima uchun mavjudligini allaqachon taxmin qilgan. Internet semantikasining tavsifi birinchi bobda berilgan ontologiya ta'rifiga to'liq mos keladi. Barcha veb-ontologiyalar bir xil semantik ma'noga ega, bu esa ushbu hujjatlarni yozish maqsadini - Internetda ma'lumotni nashr qilishni belgilaydi. Shu nuqtai nazardan, muayyan veb-hujjatlar va xizmatlarning barcha ontologiyalari yagona ontologiyaga - Web ontologiyasiga birlashtirilishi mumkin.

OWL tili, uning mualliflariga ko'ra, asosan ushbu ontologiyani tavsiflash uchun mo'ljallangan. Albatta, OWL dan boshqa ontologiyalarni tavsiflash uchun ham foydalanish mumkin, buni keyinroq ko'rib chiqamiz, hatto maxsus Protege muharriri ham ishlab chiqilgan bo'lib, u OWL tilida tavsiflangan ontologiyalar mazmuni haqida ma'lumotni qulay tarzda terish imkonini beradi.

XML tili

XML (Extensible Markup Language) Standart Generalized Markup Language (SGML) dialekti bo'lib, u o'z navbatida 1969 yilda IBM Generalized Markup Language (GML) vorisi hisoblanadi (Generalized Markup Language). SGML - bu hujjatlar uchun belgilash tillarini belgilashingiz mumkin bo'lgan meta-til. Dastlab SGML AQSH hukumati hujjatlarini kompyuterlar tomonidan o'qilishi uchun belgilash uchun yaratilgan. Shunday qilib, SGML Qo'shma Shtatlardagi hukumat va aerokosmik agentliklar tomonidan qo'llanilgan. SGML ning asosiy kamchiligi uning murakkabligidir. Shuning uchun bo'lsa kerak, uning soddalashtirilgan dialekti XML tili mashhur bo'ldi. Keling, ushbu tilda hujjatlarni qurish tamoyillarini ko'rib chiqaylik. XML tilining asosiy elementi bu **teg** deb ataladigan narsa - hujjatni belgilash uchun bevosita foydalaniladigan belgilar ketma-ketligi. Teg burchakli qavs ichiga olingan nomdir, masalan:

<name>. Boshlanish yorlig'i va tugatish tegi o'rtasida farqlanadi. Yakunlovchi teg o'zining boshlang'ich tegi bilan bir xil nomga ega bo'lishi kerak, lekin nom oldidan slash belgisi bo'lishi kerak. Masalan:

```
<name> teg kontenti </name>
```

Tegning kontenti odatda oddiy matn (bo'sh bo'lishi mumkin), lekin u murakkabroq bo'lishi ham mumkin. Boshlash teglari atributlarga ega bo'lishi mumkin, ular shakl juftlari (nom, matn) bo'lib, matnning belgilangan maydoni uchun qo'shimcha ma'lumotlarni belgilash uchun ishlatiladi. Masalan,

```
<termdef id="dt-dog" term="dog">
```

Teglar bir-birining ichiga joylashtirilishi mumkin, ya'ni bir tegning mazmuni boshqasini o'z ichiga olishi mumkin. Shunday qilib, murakkab hujjat tuzilmalarini aniqlash mumkin. Asosiy talab shundaki, teglar butunlay bir-birining ichiga joylashtirilishi kerak, bu yerda bir misol:

```
<tag1> <tag2> </tag1> </tag2>
```

Ushbu tuzilma to'g'ri emas, teglar o'rnatilmagan, lekin bir-birini to'ldirib turadi. To'g'ri tuzilishga misol:

```
<tag1> <tag2> </tag2> </tag1>
```

Bundan tashqari, hujjatning teglari daraxt tuzilishini shakllantirishi kerak, ya'ni butun hujjat uchun bitta asosiy teg bo'lishi kerak, unda belgilash matni va boshqa barcha teglar joylashtirilgan. Shunday qilib, XML hujjati daraxt bilan ifodalanadi, uning elementlari teglar va ular belgilaydigan matn bo'limlari, daraxtning tepasi hujjatning asosiy tegi bilan belgilanadi. Quyida foydalanuvchilar haqidagi ma'lumotlarni o'z ichiga olgan belgilash hujjatiga misol keltirilgan:

```
<user-list>
```

```
<user>
```

```
<id>1</id>
```

```
<fname>Ivan</fname>
```

```
<sname>Brusenko</sname>
```

```
<age>39</age>
```

```
<jins>Erkak</sex>
```

```
</user>
```

```
<user>
```

```
<id>2</id>
```

```
<fname>Shamil</fname>
```

```
<sname>Galimov</sname>
```

```

    <age>37</age>
    <jins>Erkak</sex>
  </user>
</user>
<id>3</id>
<fname>Tatyana</fname>
<sname>Volkova</sname>
<age>23</age>
<jins>Ayol</sex>
</user>
</user-list>

```

Muayyan foydalanuvchi haqidagi ma'lumot atributlar yordamida ham o'rnatilishi mumkin:

```

<user-list>
  <user id="1" fname="Ivan" sname="Brusenko"
    age="39" sex="Erkak"/>
  <user id="2" fname="Shamil" sname="Galimov"
    age="37" sex="Erkak"/>
  <user id="3" fname="Tatyana" sname="Volkova"
    age="23" sex="Ayol"/>
</user-list>

```

Bu yerda tegning mazmuni bo'sh bo'lgani uchun hech qanday o'rab turgan teglar ishlatilmaydi, bu yopilish burchagi qavs oldiga qo'yilgan to'g'ridan-to'g'ri chiziq bilan ko'rsatiladi.

Sintaksisi yuqorida sanab o'tilgan asosiy talablarga javob beradigan XML hujjatlari yaxshi tuzilgan deb ataladi. Hujjat yorlig'i daraxti turini belgilash orqali hujjat tuzilishini qo'shimcha ravishda cheklashingiz mumkin. XML SGML dan buni amalga oshirishning bir usulini meros qilib oladi. Usul shundan iboratki, hujjatning tuzilishi uning boshida DTD (Hujjat turini aniqlash) shaklida o'rnatiladi. DTDning o'ziga xos sintaksisi bor, biz bu yerda tavsiflamaymiz, biz faqat tavsifiy misol keltiramiz:

```

<!DOCTYPE user-list [
  <!ELEMENT user-list (user*)>
  <!ELEMENT user (id, fname, sname, age, sex)>
  <!ELEMENT id (#PCDATA)>
  <!ELEMENT fname (#PCDATA)>
  <!ELEMENT sname (#PCDATA)>

```



```

<!ELEMENT age (#PCDATA)>
<!ELEMENT jins (#PCDATA) ]>
<user-list> <user>
  <id>1</id>
  <fname>Ivan</fname>
  <sname>Brusenko</sname>
  <age>39</age>
  <jins>Erkak</sex>
</user>
<user>
  <id>2</id>
  <fname>Shamil</fname>
  <sname>Gamilov</sname>
  <age>37</age>
  <jins>Erkak</sex>
</user>
<user>
  <id>3</id>
  <fname>Tatyana</fname>
  <sname>Volkova</sname>
  <age>23</age>
  <jins>Ayol</sex>
</user>
</user-list>

```

Hujjat turi ta'rifi uning asosiy tegi nol yoki undan ortiq foydalanuvchi teglarini o'z ichiga olgan "user-list" tegi bo'lishi kerakligini aytadi. Har bir foydalanuvchi tegida to'rtta teg ("fname", "sname", "age", "sex") va aniq belgilangan ketma-ketlikda bo'lishi kerak. Uning strukturasi ta'rifiga mos keladigan XML hujjati "valid" deb nomlanadi.

Garchi dastlab XML hujjatlarni belgilash uchun yaratilgan bo'lsa-da, endi bu til tubdan boshqacha tarzda qo'llaniladi. XML hujjatlari hozirda bir kompyuterdan ikkinchisiga o'tkazilishi kerak bo'lgan ma'lumotlarning matnli ko'rinishi sifatida keng qo'llaniladi. Boshqacha qilib aytganda, agar ilgari XML kompyuterlar uchun tushunarli bo'lgan hujjatlarni belgilash uchun ishlatilgan bo'lsa, endi XML allaqachon tuzilgan ma'lumotlarni ketma-ket (matn) shaklga aylantirish uchun ishlatiladi va aksincha. Tildan foydalanishning bu

yangi sifati unga yangi talablarni qo'yadi. Shunday qilib, masalan, XML hujjatlarida raqamlarni yozish formatini juda ehtiyotkorlik bilan o'rnatish kerak. Axir, raqamli ma'lumotlar matnga aylantiriladi va aksincha, hech qanday yo'qotish bo'lmasligi kerak. XML ning SGML versiyasi bu imkoniyatlarni ta'minlamaydi. Shu sababli, XML tili sezilarli darajada qayta ishlandi, til turlari, har xil turdagi ma'lumotlarni ko'rsatish uchun turli formatlar va ma'lumotlarni uzatish uchun zarur bo'lgan boshqa xususiyatlar paydo bo'ldi. Shuningdek, DTDlar yordamida hujjat strukturasi aniqlash XML sxemasi deb ataladigan mosroq usul bilan almashtirildi. Sxema - berilgan XML hujjatida uzatiladigan ma'lumotlar strukturasi ta'rif va bu ta'rif XMLda ko'rsatilgan. Hujjat sxemasida yangi turlar aniqlanishi mumkin, siz XML sxemasining o'zida aniqlanganlardan foydalanishingiz mumkin. Diagrammalar Internetning turli joylarida saqlanishi va kerak bo'lganda yuklab olinishi mumkin. Bu yerda foydalanuvchi ma'lumotlarini uzatish uchun ishlatiladigan XML hujjatining to'liq ishchi namunasi:

```
<?xml version="1.1" encoding="UTF-8"?>
<xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="user-list" type="User-list"/>
<xs:complexType name="User-list">
<xs:sequence>
<xs:element name="user" type="User"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="User">
<xs:sequence>
<xs:element name="id" type="xs:decimal"/>
<xs:element name="fname" type="xs:string"/>
<xs:element name="sname" type="xs:string"/>
<xs:element name="age" type="xs:decimal"/>
<xs:element name="sex" type="xs:string"/>
</xs:sequence>
</xs:complexType></xs:schema>
</xs:schema>
<user-list>
<user>
```

```

<id>1</id>
<fname>Ivan</fname>
<sname>Brusenko</sname>
<age>39</age>
<jins>Erkak</sex>
</user>
<user>
<id>2</id>
<fname>Shamil</fname>
<sname>Gamilov</sname>
<age>37</age>
<jins>Erkak</sex>
</user>
<user>
<id>3</id>
<fname>Tatyana</fname>
<sname>Volkova</sname>
<age>23</age>
<jins>Ayol</sex>
</user>
</user-list>

```

RDF tili

RDF (Resource Description Framework) bu Internetda joylashgan ma'lumotlarni tavsiflash imkonini beruvchi tildir. Semantik internetda ular ba'zi Web ob'ektlari haqida gapirganda, ular bu ob'ektlarni resurslar deb atashadi. RDF - bunday tavsiflash uchun til. Hujjatlar semantikasining tavsiflari kompyuterlar tomonidan tushunilishi kerakligi uchun bunday o'qishni ta'minlaydigan maxsus agent dasturlarni ishlab chiqish kerak. Shuningdek, turli dasturiy ta'minot agentlari o'rtasida ma'lumot almashish imkoniyatini ta'minlash kerak. Shunday qilib, RDF nafaqat tilning o'zi, balki shu tilda yozilgan ma'lumotlarni to'liq o'qish va almashishni ta'minlash uchun zarur bo'lgan turli xil qo'shimcha dasturiy modullarni ham anglatadi. Bu haqiqat RDF tili nomida ta'kidlangan. RDF tilining asosiy elementi - triplet yoki **Uchlik** - bu uchta ob'ektning yig'indisi:

1. *Mavzu.*
2. *Ob'ekt.*
3. *Predikat.*

Predikatlar ko'pincha xususiyatlar deb ham ataladi. **Uchlik** shuningdek, tugun-yoy-tugun munosabatlari ko'rinishidagi grafik tavsifga ega:

Subyekt – Predikat – Obyekt

Uchlik qanday ma'lumotlarni saqlaydi? Bu uning mazmuniga bog'liq. Misol uchun, agar siz XML bo'limidagi misolda bo'lgani kabi foydalanuvchilar haqidagi ma'lumotlarni saqlashingiz kerak bo'lsa, u holda **triplet** foydalanuvchi atributi uchun ma'lum qiymatni ifodalashi mumkin. Atribut nomi predikat vazifasini bajaradi, sub'ekt "foydalanuvchi" sinfining namunasi, ob'ekt esa xususiyatning o'ziga xos qiymatidir. Bizning misolimizda "id" atributi "foydalanuvchi" sinfining namunasi uchun noyob identifikator sifatida ishlatilishi mumkin. Shunday qilib, uchlik (fname, 1, "Ivan") "foydalanuvchi" sinfining "1" identifikatori bilan misolning "fname" atributining o'ziga xos qiymatini ifodalaydi. Darhaqiqat, sinflarni uchlik to'plamiga tarjima qilish avtomatik ravishda amalga oshiriladi, biz RDF tilining kengaytmasini o'rganganimizda ushbu mexanizmni batafsilroq ko'rib chiqamiz - RDF sxemasi.

Matematik nuqtai nazardan, triplet qandaydir ikkilik munosabatlarning misolidir. Umuman olganda, nisbat elementlar ketma-ketliklari to'plami bo'lib, har bir ketma-ketlikdagi elementlar soni qat'iy bo'lib, oldindan belgilangan n natural soniga teng. Agar $n = 2$ bo'lsa, u holda munosabat ikkilik deyiladi. Masalan, "turmush qurgan juftliklar" munosabatlari shaklning ("er", "xotin") juftlik elementlari to'plamini belgilaydi. Shunday qilib, ma'lum bir triplet faqat bir juftlikdan ba'zi bir munosabatlarni belgilaydi, uning nomi uchinchi parametr sifatida ham ko'rsatiladi. To'liq tavsif uchun munosabatlarni belgilashdan tashqari, ularning mazmuniga ham cheklovlar qo'yish kerak. Masalan, "er-xotinlar" munosabatlari uchun bu munosabatlarning har bir juftligining birinchi elementi erkak, ikkinchisi esa ayol bo'lishi kerakligini aniqlashtirish kerak. Bunday cheklovni ko'rsatish uchun "erkak" va "ayol" tushunchalarini kiritish kerak, keyin esa, agar shaklning uchligi bo'lsa ("ism 1", "ism 2", "turmush qurgan er-xotinlar"), keyin "ism 1" tushunchasining bir misol bo'lishi kerak "erkak" va shaxs" nomi 2 "- tushunchasi bir misol" ayol ". Bu gaplar deb ataladigan so'zlar orqali amalga oshiriladi. Bayonotlar uchun o'zgaruvchilar, mantiqiy operatsiyalar, masalan, mantiqiy "yoki" (dizyunksiya), mantiqiy "va" (bog'lanish),

mantiqiy oqibat (implikatsiya), shuningdek mavjudlik va universallik kvantlarini o'z ichiga olgan til mavjud bo'lib, ular cheklash imkonini beradi. bayonot doirasi. Masalan, yuqoridagi cheklovni quyidagicha yozish mumkin ("er-xotinlar" munosabatlarini F , "erkak" va "ayol" munosabatlarini mos ravishda M va W deb belgilaymiz.): $\forall x, y F(x, y) \Rightarrow M(x) \wedge W(y)$ - "har qanday ob'ektlar uchun x va y , agar $F(x, y)$ bajarilgan bo'lsa, u holda $M(x, y)$ bajarilgan bo'lsa, u holda $M(x)$ va ulga teng bo'ladi." RDF tili tavsiflovchi mantiqning matematik apparatiga asoslangan; keling, ushbu mantiqni batafsil ko'rib chiqaylik.

Tavsiflovchi mantiq

Tavsiflovchi mantiq (Description Logic - DL) semantik tarmoqlar va kadrlar formalizmalariga asoslanadi, lekin matematik mantiq apparatidan foydalanadi. Matematik mantiqda sintaksis va semantikaga aniq bo'linish mavjud. Sintaksis ma'lum bir mantiqiy tizim dunyosining elementlari haqida turli xil bayonotlar yoziladigan tilni belgilaydi. Semantika tavsiflangan dunyoning berilgan cheklovlarni qondiradigan qismini belgilaydi. Bunday qismlar bir nechta va hatto cheksiz ko'p bo'lishi mumkin. Dunyoning har bir bunday qismi berilgan mantiqiy tizimning modeli deb ataladi. Ular ko'pincha tavsiflovchi mantiq haqida gapirib, o'ziga xos tavsiflovchi mantiqni dunyoni tavsiflaydigan ma'lum bir mantiqiy tizim sifatida nazarda tutadi. Shuning uchun o'quvchi ma'lum turdagi mantiqiy tizimlarni o'rganuvchi fan sifatida tavsiflovchi mantiq tushunchalarini va sintaksisi ma'lum cheklovlarni qondiradigan aniq mantiqiy tizim sifatida tavsiflovchi mantiq tushunchalarini farqlashi kerak. Keling, tavsiflovchi mantiqning sintaksisi va semantikasiga qo'yilgan cheklovlarni tavsiflaymiz.

Sintaksis. Har qanday tavsiflovchi mantiqning tili quyidagi elementlardan iborat:

- Tushunchalar nomlarini bildiruvchi unar predikat belgilar to'plami.*
- Rol nomlarini bildiruvchi ko'plab ikkilik predikat belgilari.*
- Kontseptsiya va rollarga asoslangan konstruktorlar yordamida kontseptsiya atamalarining rekursiv ta'rifi.*

Tushunchalar ularga tegishli bo'lgan ob'ektlar to'plamini bildiradi, ya'ni bular birinchi bobning terminologiyasidagi sinflardir.

Rollar tushunchalar orasidagi munosabatlarni belgilaydi. Terminlar konstruktorlari ham birinchi tartibli mantiq operatsiyalari, ya'ni yuqoridagi birikma, dis'yunksiya, universallik va mavjudlik cheklovlari va boshqalar, ham rollarga cheklovlar o'rnatuvchi operatsiyalar, ya'ni ikkilik munosabatlarga. Ikkilik munosabatni teskari o'zgartirish talabi, ya'ni uning simmetriyasi ($\forall x, y R(x, y) \Rightarrow R(y, x)$), ikkilik munosabatning o'tish qobiliyatiga bo'lgan talab ($\forall x, y, zx, (zx) (zR) \Rightarrow R(x, z)$) va boshqalar. Misol uchun, XML bo'limida muhokama qilingan foydalanuvchi misolida, unar predikat belgilar to'plami $UP = \{user\}$ bo'ladi, ya'ni "foydalanuvchi" yagona tushunchasining nomini o'z ichiga oladi. Ikkilik predikat belgilar to'plami $BP = \{id, sname, fname, age, sex\}$.

Semantika. Matematik mantiqda ma'lum bir to'plam odatda mantiqiy tizimning modeli sifatida ishlatiladi, keling, uni M deb ataymiz. Unar predikat belgilar to'plamining har bir elementiga ushbu to'plamda ma'lum bir unar munosabat, ikkilik predikat belgilar to'plamining har bir elementiga ikkilik munosabat beriladi. Boshqacha qilib aytganda, har bir kontsepsiya nomi ma'lum bir sinfga - to'plamning kichik to'plamiga M va har bir rol nomiga - to'plamdagi ikkilik munosabatga mos keladi. Agar munosabatlarga qandaydir cheklovlar qo'yilgan bo'lsa, unda ular barcha munosabatlarda bajarilishi kerak. M to'plamdagi nomlar va munosabatlar o'rtasidagi har bir bunday yozishma (biz uni I bilan belgilaymiz) berilgan tavsifiy mantiq modeli yoki uning talqini deb ataladi. Foydalanuvchilar haqidagi misolimizda biz quyidagi talqinga egamiz:

- $M = \{ "Ivan", "Brusenko", "Shamil", "Galimov", "Tatyana", "Volkova", "39", "37", "23", "1", "2", "3", "Erkak", "Ayol" \}$.
- $I(foydalanuvchi) = \{ ("1"), ("2"), ("3") \}$.
- $I(id) = \{ ("1", "1"), ("2", "2"), ("3", "3") \}$.
- $I(fname) = \{ ("1", "Ivan"), ("2", "Shamil"), ("3", "Tatyana") \}$.
- $I(sname) = \{ ("1", "Brusenko"), ("2", "Galimov"), ("3", "Volkova") \}$.
- $I(yosh) = \{ ("1", "39"), ("2", "37"), ("3", "23") \}$.
- $I(jinsiy) = \{ ("1", "Erkak"), ("2", "Erkak"), ("3", "Ayol") \}$.

Ba'zi to'plamlarda mantiqiy tizimni talqin qilish klassik yondashuvdir, ammo RDF grafik talqinidan ham foydalanadi. Ushbu yondashuv haqida biroz keyinroq batafsil gaplashamiz.

Tavsiflovchi mantiqda terminologik deb ataladigan komponent - **TBox** (terminologik quti) va hukmlar komponenti - **ABox** (tasdiqlash qutisi) o'rtasida farqlanadi. **TBox** kontseptsiya ierarxiyasi haqidagi bayonotlarni o'z ichiga oladi, ya'ni tushunchalar o'rtasidagi munosabatlarni belgilaydi va **ABox**da shaxslar va tushunchalar o'rtasidagi munosabatlarni tavsiflovchi bayonotlar mavjud. Masalan, "har bir foydalanuvchi – bu shaxs" iborasi "xodim" va "odam" tushunchalari o'rtasidagi munosabatni belgilaydi va shuning uchun **TBox** to'plamiga tegishlidir. "Ivan - foydalanuvchi" iborasi "Ivan" individual va "shaxs" tushunchasi o'rtasidagi munosabatni belgilaydi va **ABox** to'plamiga tegishli. Ta'riflovchi mantiqda **TBox** to'plamining elementlari faqat unar predikatlar (tushunchalar) uchun o'rnatilgan cheklovlardir; bu ularning **ABox** to'plamining bayonotlaridan farqidir. Agar tavsiflovchi mantiq modellari bo'yicha xulosa chiqarish protsedurasini qurish imkoniyatini ko'rib chiqsak, **TBox** va **ABox** bo'yicha bayonotlarni farqlash foydali bo'ladi. **TBox** bayonotlari "tasniflash" xususiyatlarini belgilaydi va **ABox** bayonotlari shartli ravishda "namuna tekshiruv" deb nomlanishi mumkin bo'lgan xususiyatlarni belgilaydi. Ushbu to'plamlar bo'yicha xulosalar ishlashda sezilarli darajada farq qilishi mumkin, shuning uchun har bir komponent uchun alohida xulosa algoritmlarini amalga oshirish mantiqan. Bundan tashqari, **TBox** to'plamidan ushbu tavsiflovchi mantiqni amalga oshiradigan dasturiy ta'minot tizimiga interfeys sifatida foydalanish mumkin. Veb-semantikada qo'llaniladigan turli tavsiflovchi mantiqlar uchun asosiy til **ALC** (To'ldiruvchilar bilan atributiv til) hisoblanadi. **ALC** uchta komponentdan iborat:

- *Asosiy sinf nomlari to'plami (NC), shuningdek ikkita maxsus sinf: universal sinf / va bo'sh sinf \perp .*
- *Property nomlari to'plami (NR).*
- *Ob'ekt nomlari to'plami (NI).*

Tayanch sinflarning nomlari asosida murakkabroq sinflarni qurish mumkin. Konyunksiya, diszyunksiya, inkor qilish, qo'shish va ekzistensial va universallik kvantorlaridan foydalanishning mantiqiy amallari yangi sinflar konstruktori vazifasini bajaradi. Misol uchun, agar asosiy sinflar Ona va Ota mavjud bo'lsa, unda mos keladigan predikatlarni ajratish operatsiyasidan foydalanib, siz ota-onaning yangi sinfini belgilashingiz mumkin. Bundan tashqari, sinf-kichik sinf munosabati, sinf misoli va xususiyat misoli ham mavjud. **ALC**

semantikasi har bir sinfga to'plam va har bir munosabatga juft elementlar to'plamini tayinlaydigan sharhlovchi funktsiya bilan belgilanadi. Shubhasiz, **ALC** tavsiflovchi mantiqning o'ziga xos turi bo'lib, unda yuqoridagi nomlash aksiomalari va pastki sinf munosabatlari terminologik komponent (**TBox**) va yangi sinflarni qurish va misollarni taqsimlash aksiomalari hukm komponenti (**ABox**) sifatida ishlaydi. **ALC** tili asosida, yangi aksiomalarni qo'shish orqali, **OWL** tilining turli dialektlarida qo'llaniladigan cheklashlar, tavsiflovchi mantiqlar amalga oshiriladi, ular quyida batafsil muhokama qilinadi.

Tavsiflovchi mantiqlarni tavsiflashda **SHOIN** (\mathcal{D}) yoki **SROIQ** (\mathcal{D}) kabi turli qisqartmalar qo'llaniladi. Keling, ushbu qisqartmalarning ma'nosini tushunib olaylik:

- *Tavsifiy mantiq nomidagi **S** harfi bu mantiq **ALC** tili sifatida amalga oshirilganligini bildiradi, bunda qo'shimcha ravishda munosabatlar uchun tranzitivlik aksiomasini kiritish mumkin.*
- ***SH** mantiqiy **S** bo'lib, munosabatlar ierarxiyasini saqlaydi.*
- ***SHI** – **SH** mantiqidir, unda berilgan munosabatlar uchun teskari munosabatlarni aniqlash mumkin.*
- ***SHIF** – funktsional munosabatlarni aniqlash qobiliyatiga ega **SHI** mantig'idir.*
- ***SHIN** mantig'i **SHI** bo'lib, unda munosabatlar qiymatlari to'plamining o'lchamiga cheklovlar o'rnatish mumkin. Ushbu cheklovning ingliz tilidagi nomi "malakali bo'lmagan raqam cheklovlari" dir.*
- ***SHIQ** – bu **SHI** mantig'i bo'lib, unda siz munosabatlarning qiymatlar to'plamining o'lchamiga cheklovlar o'rnatishingiz va ma'lum miqdordagi elementlar bilan qandaydir munosabat bilan bog'liq bo'lgan elementlardan kichik sinflarni belgilashingiz mumkin (malakali son cheklovlari). Misol uchun, bir nechta avtomobilga ega bo'lgan shaxslar sinfini quyidagicha belgilash mumkin:*
- ***SHOIQ** mantig'i **SHIQ** bo'lib, deb atalmishni o'rnatish qobiliyati bilan kengaytirilgan. nominatsiyalar. Denominatsiyalar sanab ko'rinishidagi sinflarni belgilashdir. **RDFS** tavsifida ushbu toifadagi shaxslarni (o'ziga xos qit'alar) ro'yxatga olish orqali Kontinent sinfini aniqlash misolidan foydalaniladi.*

- ***SROIQ** mantig'i **SHOIQ** bo'lib, unda ularning tarkibi orqali munosabatlar ierarxiyasini aniqlash mumkin.*
- ***SROIQ**(\mathcal{D}) – **SROIQ** mantig'ini ifodalaydi, unga mantiqiy deb ataladi. ma'lumotlar turlari va qiymatlari ma'lumotlar turlari bo'lishi mumkin bo'lgan munosabatlar. Ma'lumotlar turlari butun sonlar, haqiqiy sonlar yoki satrlar kabi XMLda belgilangan turlardir.*

RDF sintaksisi

RDF internetning turli qismlarida taqsimlangan resurslarni tavsiflash uchun ishlatilishi kerakligi sababli, RDF grafigining tugunlari va qirralari nomlarini aniqlash muammosini qandaydir tarzda hal qilish kerak, ya'ni uchlik elementlari. Buning uchun standart yondashuv qo'llaniladi: har bir element yagona resurs identifikatori (URI) yordamida tavsiflanadi. Odatda, URI bu yoki Internetdagi berilgan resursning joylashuvi haqidagi ma'lumotlarni o'z ichiga olgan URL (Uniform Resource Locator) yoki nomlar maydonidagi berilgan resursni identifikatsiya qiluvchi URN (Uniform Resource Name). Nomlar maydoni shunchaki nomlar to'plamidir va bu nomlar Internetda yagona bo'lishini ta'minlash uchun ishlatiladi. Misol uchun, agar ikkita RDF grafigini (uchlik to'plamlari) birlashtirish muammosi mavjud bo'lsa va bir vaqtning o'zida har bir grafikda mahalliy "Ivan" nomi bilan o'z tugunidan foydalansa, siz muammoni hal qilishingiz mumkin. Grafiklar birligidagi ushbu tugunlarni ikki usulda aniqlash:

1. Internetdagi RDF grafigining URL manzilidan foydalanib, tugunlarning har biriga noyob nom bering. Misol uchun, agar birinchi grafikning URL manzili <http://www.example.com/graph1#> va ikkinchisining URL manzili <http://www.example.com/graph2#> bo'lsa, Ivan tugunining URI'si birinchi ustunda <http://www.example.com/graph1#Ivan>, ikkinchi ustundagi "Ivan" tugunining URI - <http://www.example.com/graph2#Ivan>.

2. Grafiklarning har biri uchun o'z nom maydonini belgilang va keyin "Ivan" mahalliy nomi bilan tugunlarning nomlari grafiklar birligida yagona bo'ladi, chunki turli nom maydonlarida bo'ladi. Masalan, birinchi grafik uchun uning nom maydonini quyidagicha o'rnating: `xmlns: graph1 = "http://www.example.com/graph1#"`, ikkinchisi uchun esa

shunday: xmlns: graph2 = "http://www. example.com/ graph2 #". Keyin, birinchi grafikdagi "Ivan" tugunining to'liq nomi grafik1 bo'ladi: Ivan, ikkinchi grafikda2: Ivan.

URI'lar, shuningdek, turli RDF hujjatlaridan bir xil veb-ob'ektlarga murojaat qilish imkonini beradi. Semantik Internet uchta standart nom maydonidan foydalanadi:

- **rdf.** Ushbu nom maydoni RDFda ishlatiladigan nomlarni belgilaydi. Bu nom uchun URI: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.

- **rdfs.** Bu yerda RDF sxemasida ishlatiladigan nomlar o'rnatiladi. Uning URI manzili <http://www.w3.org/2000/01/rdf-schema#>.

- **OWL.** OWL da ishlatiladigan nomlarni tavsiflaydi. Bu nom uchun URI: <http://www.w3.org/2002/07/owl#>. RDFda ishlatiladigan eng keng tarqalgan nomlar *rdf: type*, *rdf: isInstanceOf* va *rdf: Property*.

Keling, ular bizning misolimizda foydalanuvchilar jadvali bilan qanday qo'llanilishini tushuntiramiz. Yuqorida aytib o'tilganidek, siz RDF-dan foydalanib, jadval mazmunini quyidagi universal algoritm bilan tavsiflashingiz mumkin:

- Jadvalning har bir satrini shu qator uchun o'ziga xos identifikatori bilan ifodalang. Odatda, bu faqat jadvaldagi berilgan qatorning soni.

- Har bir ustun nomini *rdf* sifatida ko'rsating: *Property*.

- Har bir jadval nomini *rdf: type* sifatida ko'rsating.

Bizning misolimizda quyidagi jadval mavjud:

Ustun nomlarini *rdf* sifatida ta'riflaymiz: *Property*, ya'ni deylik, bu nomlar predikat vazifasini bajaradi. Barcha nomlar **usr** nom maydonida aniqlangan deb taxmin qilinadi.

(usr:id,rdf:type,rdf:Property)

(usr:fname,rdf:type,rdf:Property)

(usr:sname,rdf:type,rdf:Property)

(usr:age,rdf:type,rdf:Property)

(usr:sex,rdf:type,rdf:Property)

Endi har bir satr uchun “foydalanuvchi” sinfining “nasolini” aniqlaymiz:

(usr:user1,rdf:type,usr:user)

(usr:user2,rdf:type,usr:user)

(usr:user3,rdf:type,usr:user)

Bu yerda biz usr birlashtirilgan misol nomidan foydalanamiz: usern, bu yerda n - satr identifikatori. Keling, sinflarning xususiyatlarini sanab o'tamiz.

(usr:user1,usr:id,1)
(usr:user1,usr:fname,Ivan)
(usr:user1,usr:sname,Brusenko)
(usr:user1,usr:age,39)
(usr:user1,usr:jins,erkak)
(usr:user2,usr:id,2)
(usr:user2,usr:fname,Shamil)
(usr:user2,usr:sname,Galimov)
(usr:user2,usr:age,37)
(usr:user2,usr:jins,erkak)
(usr:user3,usr:id,3)
(usr:user3,usr:fname,Tatyana)
(usr:user3,usr:sname,Volkova)
(usr:user3,usr:age,23)
(usr:user3,usr:jins,ayol)

E'tibor bering, bu yerda **uchlik** tavsifining boshqa tartibi ishlatilgan, ya'ni: (mavzu, predikat, ob'ekt). Bu tartib standart belgi sifatida ishlatiladi va biz keyinroq unga amal qilamiz. XML4 hujjatlaridan tashqari **uchlik**larni tavsiflash uchun ikkita standart belgi qo'llaniladi. Keling, ularni tavsiflab beraylik.

N-Triple 5 yozuvi. Ushbu belgida tripletning har bir elementi burchakli qavslar ichiga olingan URI bilan ifodalanadi. Elementlar tartibda sub'ekt, predikat, ob'ekt va nuqta bilan tugaydi. Misol uchun, agar yuqoridagi misoldagi nomlar URL = <http://www.example.org/user-list#> da joylashgan bo'lsa, **uchlik** (usr: user1, rdf: type, usr: user) shunday ko'rinadi:

<http://www.example.org/user-list#fname>
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://www.example.org/user-list#user>.

RDF yozuvi 3 (N3). Bu prefikslar deb ataladigan va @prefix tugmalar birikmasi bilan belgilangan mahalliy URI sinonimlarini aniqlash imkonini beruvchi yanada ixcham belgidir. RDF uchliklarini yozishning bu usuli odamlar uchun maxsus ishlab chiqilgan va XML RDFga qaraganda inson uchun qulayroq belgidir. Uchlik

elementlarining nomlari burchakli qavslarsiz "mavzu, predikat, ob'ekt" tartibida yoziladi va nuqta bilan tugaydi. Agar bir mavzu bilan bir qatorda bir nechta uchlik yozilsa, u holda uchlik oxirida, agar oxirgi bo'lmasa, nuqta-vergul, oxirgi uchlik oxirida esa nuqta qo'yiladi. Bunday holda, mavzu nomi faqat birinchi uchlasiga qo'yiladi, qolganlarida u avtomatik ravishda avvalgisidan almashtiriladi. Uchtasi quyidagicha yoziladi:

```
@prefix usr <http://www.example.org/user-list#>
@prefix rdf <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
    usr:user1 rdf:type usr:user
    usr:user2 rdf:type usr:user
    usr:user3 rdf:type usr:user
    usr:id rdf:type rdf:Property .
    usr:fname rdf:type rdf:Property .
    usr:sname rdf:type rdf:Property .
    usr:age rdf:type rdf:Property .
    usr:sex rdf:type rdf:Property .
    usr:user2 usr:id "1" ;
    usr:fname "Ivan" ;
    usr:sname "Brusenko" ;
    usr:age "39" ;
    usr:sex "erkak" .
    usr:user2 usr:id "3" ;
    usr:fname "Shamil" ;
    usr:sname "Galimov" ;
    usr:age "37" ;
    usr:sex "erkak" .
    usr:user2 usr:id "2" ;
    usr:fname "Tatyana" ;
    usr:sname "Volkova" ;
    usr:age "23" ;
    usr:jins "Ayol" .
```

XML hujjatlarida uchliklarni yozish yuqorida muhokama qilingan belgilarga qaraganda biroz boshqacha tarzda amalga oshiriladi. Keling, yuqoridagi foydalanuvchilar bilan misoldagi RDF grafigi qanday yozilishini ko'rib chiqaylik. user1, user2 va user3 sub'ektlari rdf: about maxsus tegida qayd etilgan. Ushbu mavzularning har biri bilan uchlik ushbu tegning bolalari sifatida yoziladi. Har bir

bola tegi predikat nomi, uning mazmuni esa ob'ektdir. Barcha ta'riflar rdf: RDF ildiz tegida berilishi kerak. XML hujjatlaridagi RDF sintaksisi haqida batafsil ma'lumot uchun.

```
<rdf:RDF>
  xmlns:usr="http://www.example.org/user-list#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  <usr:user rdf:about="http://www.example.org/user-
    list#user1">
    <usr:id>1</usr:id>
    <usr:fname>Ivan</usr:fname>
    <usr:sname>Brusenko</usr:sname>
    <usr:age>39</usr:age>
    <usr:jins>erkak</usr:sex>
  </usr:user>
  <usr:user rdf:about="http://www.example.org/user-
    list#user2">
    <usr:id>2</usr:id>
    <usr:fname>Shamil</usr:fname>
    <usr:sname>Galimov</usr:sname>
    <usr:age>37</usr:age>
    <usr:sex>erkak</usr:sex> </usr:user>
  <usr:user rdf:about="http://www.example.org/user-
    list#user3">
    <usr:id>3</usr:id>
    <usr:fname>Tatyana</usr:fname>
    <usr:sname>Volkova</usr:sname>
    <usr:age>23</usr:age>
    <usr:sex>ayol</usr:sex>
  </usr:user>
</rdf:RDF>
```

Xulosa qilib aytishimiz mumkinki, RDF grafiklari "mavzu, predikat, ob'ekt" ko'rinishidagi uchlik to'plamlar bo'lib, ularning elementlari URIlar bilan belgilangan resurslar yoki literal ma'lumotlardir (literallar faqat ob'ektlar sifatida harakat qilishi mumkin). Ba'zan elementlar sifatida Internetda vakili bo'lmagan manbadan foydalaniladi, bu RDF grafigining bo'sh tugunidir. Bo'sh tugunlar N3 yozuvida savol belgisi bilan belgilanadi. Nima uchun bo'sh tugunlar kerak? Keling, bir misolni ko'rib chiqaylik. Ma'lum

bo'lsinki, Shekspirning Angliyada yashagan va asrlar davomida nomi saqlanib qolmagan sevgilisi bo'lgan, lekin unga bo'lgan muhabbat Shekspirni 78-sonatasini yozishga ilhomlantirgan. Bu haqiqatni quyidagicha ifodalash mumkin:

*lit:Mistress1 rdf:type bio:Woman;
bio:LivedIn geo:England .*

lit:Sonnet78 lit:hasInspiration lit:Mistress1.

Ammo biz haligacha Shekspirning sevgilisi ismini bilmaganimiz uchun uning nomiga ahamiyat berishning hojati yo'q: *lit:Mistress1*. Bu haqiqatni quyidagicha ifodalash yaxshiroqdir:

*?rdf:type bio:Woman;
bio:livedIn geo:England .*

lit:Sonnet78 lit:hasInspiration? .

Bo'sh tugunga ega bo'lgan uchlik, uchlik qoniqtiradigan ba'zi ob'ektning mavjudligi faktini ifodalaydi, ya'ni mantiq nuqtai nazaridan, “*? rdf: type bio: Woman* ” $\exists x \text{ rdf: type } (x, \text{ bio: Ayol})$ deyishga teng. Agar sizda bir nechta turli bo'sh tugunlarga ega bo'lishingiz kerak bo'lsa, uni o'z ichiga olgan uchliklarni kvadrat qavslar ichiga olish orqali ma'lum bir bo'sh tugundan foydalanish kontekstini ajratib ko'rsatishingiz mumkin. Bunday holda, savol belgisi endi kerak emas, shunchaki o'tkazib yuboriladi:

*[rdf:type bio:Woman;
bio:livedIn geo:England].*

SPARQL so'rovlar tili

Ehtimol, o'quvchida RDF tilida ma'lum bir ontologiyaning tavsifi yo'naltirilgan grafik (qirralari orientatsiyaga ega bo'lgan grafik, ya'ni ular o'qlar), qirralari predikatlar bilan belgilangan va cho'qqilar - berilgan chetni uning uchlari bilan birga ifodalovchi uchlikning predmeti va ob'ekti haqida ma'lum bir farazlar paydo bo'lgandir. Bunday grafiklar uchun biz RDF ombori atamasidan foydalanamiz. Shu ma'noda, RDF saqlash maxsus tuzilgan ma'lumotlar bazasiga o'xshaydi. Relyatsion ma'lumotlar bazasidan ma'lumotlarni olish odatda maxsus SQL tilida yozilgan so'rovlar (Structured Query Language - Structured Query Language), ma'lumotlar bazasi so'rovlari yordamida amalga oshiriladi. Siz shunga o'xshash yondashuvni RDF omborlari uchun qo'llashni kutishingiz mumkin. Haqiqatan ham, ushbu monografiyani yozish vaqtida RDF ombori so'rovlar tilining ko'plab turli dialektlari mavjud. W3 Konsorsiumi o'zining standarti

sifatida shunday tillardan birini tanladi - SPARQL ("sparkle" deb talaffuz qilinadi). SPARQL rekursiv qisqartma bo'lib, SPARQL Protocol va RDF Query Language degan ma'noni anglatadi. Shubhasiz, nom SQL qisqartmasi bilan parallel bo'ladi. Keling, ushbu tilning asosiy g'oyalarini muhokama qilaylik. Oldingi bo'limda biz yaratgan foydalanuvchi ma'lumotlariga ega RDF grafigi so'rovlar uchun saqlash joyi sifatida ishlatiladi. SPARQL so'rovining asosiy elementi uchlik naqshdir. Uchlik qolip - bu uchlik bo'lib, uning ba'zi joylarida sub'ekt, predikat yoki ob'ekt o'rniga o'zgaruvchilar mavjud. O'zgaruvchilar "?" savol belgisi bilan boshlanadigan identifikatorlar bilan belgilanadi. Masalan:

*?user usr:fname "Ivan" .
usr:user2 usr:sname ?name .
?user usr:fname ?name .*

Uchlik uchun naqshlarni yozish sintaksisi N3 yozuvida uchlik yozish sintaksisiga juda o'xshaydi, har bir uchlik ham nuqta bilan tugaydi. Yuqoridagi uchlik naqshlarining har biri savol sifatida ko'rib chiqilishi mumkin:

- *Foydalanuvchi sinfining barcha nusxalarini usr bilan chop eting: fname "Ivan" ga teng.*
- *usr: user2 misoli uchun usr: sname xususiyatining qiymati qanday?*
- *usr ning barcha qiymatlarini chop eting: repozitoriydagi foydalanuvchi sinfining barcha namunalari uchun fname xususiyati.*

Ushbu savollarga javob berish uchun SPARQL so'rovlar tizimi RDF xotirasini qidiradi va o'zgaruvchilar o'rnida istalgan qiymatga ega bo'lishi mumkin bo'lgan barcha tripletlarni, boshqa joylarda esa triplet shablonining matnida ko'rsatilgan resurslarni ko'rsatadi. Shubhasiz, agar uch shablon "? Sub? Pred? Obj"ga o'xshasa, natijada barcha saqlash uchliklari ko'rsatiladi. RDF saqlash - bu grafik, shuning uchun qidiruv uchun uch shablon o'rniga grafik shablonlaridan foydalanish juda tabiiy. . Grafik shablon uchlik to'plami bo'lib, bir xil nomdagi o'zgaruvchilar shablonning barcha uchliklarida bir xil qiymatlar bilan to'ldirilishi kerak bo'lgan xususiyatga ega. SPARQL da grafik shablon jingalak qavslar ichiga olingan. Masalan, grafik shablon:

{?user usr:fname "Ivan" . ?user usr:sname ?sname . }

ismi "Ivan" bo'lgan foydalanuvchilarning barcha familiyalarini, shuningdek, `usr: fname "Ivan"` bo'lgan ob'ektlarning nomlarini ko'rsatadi. Grafik shablonlari mantiqiy ravishda birlashtirilishi mumkin. UNION kalit so'zi mantiqiy yoki sifatida ishlatiladi. Masalan, biz "Ivan" yoki "Tatyana" bo'lgan foydalanuvchilarning familiyalarini ko'rsatamiz.

```

{
  {
    {?user usr:fname "Ivan" .}
    UNION
    {?user usr:fname "Tatyana" .}
  }
  {?user usr:sname ?sname .}
}

```

RDF sxemasi

RDF sxemasi (RDF sxemasi, RDFS) RDF tilining kengaytmasi bo'lib, RDF omborlarida saqlangan ma'lumotlarning oddiy ontologiyalarini tavsiflash imkonini beradi. Ma'lumotlar bazasi sxemasi ma'lumotlar bazasi tuzilishini jadval sarlavhalari va ular o'rtasidagi munosabatlar nuqtai nazaridan tavsiflaganidek, RDF sxemasi RDF do'konining tuzilishini tavsiflash imkonini beradi. Tuzilish turlari va ular o'rtasidagi munosabatlar nuqtai nazaridan tavsiflanadi. Darhaqiqat, o'quvchi bunga biroz keyinroq ishonch hosil qilishi sababli, RDF sxemasi faqat ba'zi qo'shimcha aloqalar bilan tasniflarni tavsiflashga imkon beradi. Muloqotlarning yanada murakkab turlarini tavsiflash uchun OWL kabi kuchliroq vositalardan foydalanish kerak. Sinflar. Birinchi bobda biz sinflar o'z misollarining terilgan to'plami ekanligini aniqladik. Sinf bir tur bo'lganligi sababli, bu sinf elementlariga ega bo'lgan xususiyatlar to'plami mavjud. Bundan tashqari, bir sinf misollari to'plami boshqa sinf misollarining kichik to'plami bo'lsa, kiritiladigan tip-kichik turdagi munosabatlar mavjud. Tur-kichik tip munosabati ierarxik munosabatlar deb ham ataladi. Taksonomiya - har bir ob'ekt kamida bitta sinfga tegishli bo'lgan va sinf turlari ierarxik daraxtni tashkil etadigan tasniflashning eng oddiy shakli. Keling, RDFSda sinflar qanday aniqlanishini ko'rib chiqaylik. RDFS barcha sinflar uchun sinfni belgilaydi, u rdfs sifatida belgilanadi: `Class`. Har bir sinf rdfs namunasi sifatida ko'rsatilgan: `Class` klassi. Shu nuqtada, o'quvchi hayron bo'lishi mumkin: rdfs:

Class o'zining namunasi sifatida belgilanganmi? Bu haqiqatan ham shunday, barcha sinflar sinfi quyidagicha aniqlanadi. Birinchidan, `rdfs: Resurs` klassi ko'rsatilgan, bu har qanday RDFS ob'ektini bildiradi. `rdfs: Resurs` - bu sinf, ya'ni `rdfs` misoli: `Class`. Biroq, har bir RDFS ob'ekti `rdfs` ning namunasi: `Resurs` klassi va shuning uchun `rdfs: Sinf` `rdfs`ning namunasi: `Resurs`. Faqat sinflarda misollar bo'lishi mumkin, ular o'z navbatida `rdfs` misollariga ega: `Class` klassi. Shunday qilib, barcha sinflarning klassi rekursiv tarzda, ya'ni o'zining namunasi sifatida ko'rsatiladi. Sinf o'zining namunasi bo'lishi mumkin degan taxmin biroz g'alati. To'plamlar nazariyasida Rassellning paradoksi ma'lum bo'lib, u ushbu taxminga asoslanadi. Albatta, oddiy tasnifning xossalarini ifodalash uchun o'ziga xos misollar bo'lgan sinflardan foydalanishning hojati yo'q.

Yuqorida aytib o'tilganidek, siz `rdf: type` yordamida berilgan ob'ektni berilgan sinfning namunasi deb aytishingiz mumkin. Masalan, `usr: user` `rdf: type` `rdfs: Class` "`usr: user`" `rdfs` ning namunasi ekanligini aytadi: Barcha sinflarning sinf klassi va shuning uchun sinf. Xususiyatlari. RDFS shuningdek, `rdf` bilan belgilangan barcha xususiyatlar sinfiga ega: `Property`. Barcha xususiyatlar ushbu sinfning namunalari va uning o'zi, o'z navbatida, `rdfs: Class` sinfining namunasi. Ayrim mulkning qiymatlari ba'zi bir sinfning namunalari ekanligini aytish uchun, ya'ni `rdfs: range` xususiyati xossalar turlarini belgilash uchun ishlatiladi. `P` `rdfs: diapazon` `C` ifodasi `P` `rdf: Property` sinfining namunasi va `C` `rdfs: Class` sinfining namunasi ekanligini va predikati `P` xossasi bo'lgan tripletlarga ob'ektlar sifatida kiritilgan barcha resurslar sinf misollari ekanligini bildiradi. `C` OOPda bu holat qiymat turi deklaratsiyasiga mos keladi. Masalan, `C ++` tilining "`intage`" deklaratsiyasi; `age` xususiyatining `int` turini o'rnatadi, ya'ni yoshi xususiyatining barcha ob'ektlari `int` sinfining namunalari ekanligini aytadi. Agar `rdfs: diapazon` ma'lum bir xususiyat oladigan qiymatlar turini belgilashga imkon bersa, `rdfs: domen` xususiyati ushbu xususiyat atributi bo'lgan sinfni ko'rsatishga imkon beradi. `P` `rdfs: C` domeni ifodasi `P` `rdf: Xususiyat` sinfining namunasi va `C` `rdfs: Sinf` klassining namunasi ekanligini va predikati `P` xossasi bo'lgan tripletlarga sub'ektlar sifatida kiritilgan barcha resurslar `C` misollar sinfi ekanligini aytadi. Agar `usr: age` xossasi `usr: user` sinfining atributi bo'lsa, u holda biz bu haqda shunday aytishimiz mumkin: `usr: age` `rdfs: domen` `usr: foydalanuvchi`. Shunday qilib, `C ++` ifodasi:

```

class usr:user{
    int usr:age;
};
usr:age rdfs:range int .
usr:age rdfs:domain usr:user.

```

rdfs: subClassOf xususiyati OOPdagi merosga o'xshaydi. Ifoda:

C1 rdfs:subClassOf C2

C1 turi C2 turining pastki turi ekanligini anglatadi, ya'ni C1 sinfining har bir misoli C2 sinfining namunasi. Agar sinflarning merosxo'rliqi keng tarqalgan bo'lsa va OOPda tez-tez ishlatilsa, u holda xususiyatlarni meros qilib olish, birinchi qarashda, tushunarsiz ko'rinadi. Ammo, agar biz mulkni bir xil predikatli uchlik to'plami deb hisoblasak, hamma narsa joyiga tushadi. Bunday uchliklarning barchasi juft holda ifodalanishi mumkin (mavzu, ob'ekt), bu yerda biz yana predikatni ikkilik munosabat sifatida ko'rib chiqamiz. Agar bir xossaning juftliklari to'plami boshqa xossaning juftliklari to'plami bo'lsa, u holda mulk-kichik qiymat munosabati kiritilishi mumkin. RDFSda bu rdfs: subPropertyOf xususiyati yordamida amalga oshiriladi. Ifodasi:

P1 rdfs:subPropertyOf P2

P1 va P2 rdf ning misollari ekanligini aytadi: Xususiyat sinfi va agar uch R1 P1 R2 bo'lsa, u holda uch R1 P2 R2 bo'lishi kerak. rdfs: subPropertyOf munosabati o'tish xususiyatiga ega (har xil munosabatlar xususiyatlarini biroz keyinroq ko'rib chiqamiz).

RDFS shuningdek, konteynerlarni (rdfs dan meros bo'lib qolgan sinflar: Konteyner) va kolleksiyalarni (rdf sinfi: List) aniqlash imkonini beruvchi elementlarni ham belgilaydi, biz bu yerda ko'rib chiqmaymiz. RDF lug'atining barcha elementlari bilan batafsil tanishish uchun o'quvchi W3 konsorsiumi veb-saytidagi RDFSning rasmiy tavsifiga murojaat qilishi mumkin. Ko'pincha ikkita lug'at elementi ham ishlatiladi: rdfs: label va rdfs: comment. rdfs: label insonga qulay manba nomlarini belgilash imkonini beradi va rdfs: turli RDF sxema ta'riflariga izoh berish uchun. Bu RDFni ko'rib chiqishimizni yakunlaydi va RDFga qaraganda ko'proq ifodali kuchga ega OWLni tavsiflashga o'tadi.

OWL ideologiyasi

OWL (Web Ontology Language) ontologiyalarni tavsiflovchi til bo'lib, W3 konsorsiumi tomonidan maxsus shu maqsadda ishlab

chiqilgan. OWL DAML va OIL aralashmasidan iborat bo'lgan DAML + OIL ning qidirish mumkin bo'lgan versiyasidir. DAML (DARPA Agent Markup Language) harbiylar uchun yangi texnologiyalarni ishlab chiqish uchun tashkil etilgan AQSh Mudofaa vazirligi qoshidagi DARPA agentligi uchun ishlab chiqilgan. DAML - bu WWW da "agentlarni belgilash" uchun til; turli veb-mijozlarni tavsiflash uchun til. OIL (Ontology Inference Layer) tili yevropalik tadqiqotchilar tomonidan Web ontologiyalarini tavsiflash uchun ishlab chiqilgan. DAML + OIL ni OWL ning dastlabki chiqarilishi deb hisoblash mumkin. OWL RDF va RDFSning kengaytmasi sifatida qurilgan. Bu shuni anglatadiki, tilning asosiy sintaksisi hali ham XML, asosiy konstruktsiyasi esa RDF tripletidir. Shu nuqtai nazardan, OWLni RDFS ning kengaytirilgan versiyasi sifatida ko'rish mumkin, bu nafaqat sinflar va xususiyatlarni tavsiflash, balki ulardan foydalanishga cheklovlar o'rnatish imkonini beradi. Ta'riflovchi mantiq tilida bu shuni anglatadiki, OWL asosidagi mantiq munosabatlarni tavsiflashdan tashqari, ushbu munosabatlar o'rtasidagi munosabatlarni va ikkinchisiga nisbatan turli xil cheklovlarni belgilaydigan aksiomalarni o'z ichiga oladi.

OWL asoslari: *Sinflar*

OWL tilining asosiy elementi OWL sifatida belgilangan barcha sinflar sinfidir: Class. OWL: Sinf klassi rdfsning namunasi: Biz yuqorida muhokama qilgan sinf. Shuning uchun, har qanday OWL klassi OWLning namunasi sifatida ko'rsatilishi kerak: Sinf klassi. Misol uchun, agar biz Inson sinfini aniqlamoqchi bo'lsak, biz uchta Inson rdf ni aniqlashimiz kerak: OWL turi: XML sintaksisida shunday ko'rinishga ega bo'lgan sinf: <owl: Sinf rdf: ID = "Inson" /> OWL shuningdek ikkitadan iborat. oldindan belgilangan sinflar:

· *OWL sinfi: barcha shaxslar to'plamini bildiruvchi narsa sinfi (varlik).*

· *OWL sinfi: bo'sh to'plamni bildiruvchi hech narsa sinfi yo'q.*

Har bir OWL sinfi owlning bolasi: Thing sinfi va owlning otanasi: Hech narsa sinfi. Shunday qilib, OWLdagi meros daraxti to'liq panjara deb ataladigan narsani hosil qiladi. Odatda, ob'ektni qandaydir sinfning namunasi sifatida e'lon qilish kifoya, shunda bu shaxs avtomatik ravishda OWLning a'zosi bo'ladi: Thing sinfi.

OWL da sinf merosi rdfs yordamida belgilanadi: subclassOf qurilishi, ya'ni xuddi RDF sxemasi kabi. RDF sxemasida bo'lgani

kabi, bir sinf boshqa sinfning pastki sinfi ekanligi, bolalar sinfining barcha namunalari ota-sinfning namunalari ekanligini anglatadi.

OWL klassi va qiymat aksiomalari

OWL tili sizga sinflar va xususiyatlarning turli xususiyatlarini tavsiflash imkonini beradi, ular odatda ularning misollari o'rtasidagi munosabatlar tuzilishiga turli xil cheklovlar sifatida o'rnatiladi. Bu cheklovlar OWL da aksiomalar deb ataladigan oldindan belgilangan munosabatlar shaklida ifodalanadi. Keling, ushbu munosabatlarni batafsil ko'rib chiqaylik.

Sinf aksiomalari

Sinf aksiomalari sinf misollari bir-biri bilan bog'langan qoidalarni belgilaydi. Sinfning asosiy aksiomasi uning nomini sinf nomi sifatida e'lon qilishdir. Bu aksioma shunday nomli sinf mavjudligini aytadi. Ammo bu, albatta, sinflar orasidagi oddiy ierarxik munosabatlarni ifodalash uchun ko'pincha yetarli emas. Shuning uchun, OWL tilida yana uchta sinf aksiomasi ajralib turadi:

- RDFSda ham mavjud, `rdfs: subclassOf` aksiomasi, sinfni boshqasining kichik sinfi sifatida tavsiflaydi. Yuqorida aytib o'tilganidek, agar sinf boshqa sinfning pastki sinfi bo'lsa, uning barcha misollari ham ota-sinfning namunalari hisoblanadi.

- OWL: `ekvivalentClass` klassi ekvivalentlik aksiomasi, bu bizga ushbu sinflarning misollar to'plami mos kelishini aytishga imkon beradi. Odatda, bu sinonimlarni aniqlashning bir usuli. Misol uchun, siz Bosh direktor deb nomlangan sinfni belgilashingiz mumkin yoki siz Kompaniyaning sinf rahbari haqida ham aytishingiz mumkin. Keyin ushbu sinflarning ekvivalentligini quyidagicha belgilashingiz mumkin: Bosh direktor OWL: `ekvivalentClass` kompaniyasi rahbari. Ko'pincha ekvivalentlik aksiomasi nom bilan emas, balki qandaydir amallar bilan, masalan, ularning elementlarini sanab o'tish orqali ko'rsatilgan sinflarni nomlash usuli sifatida ham qo'llaniladi.

- Oldingi ekvivalentlik aksiomasi ma'nosida qarama-qarshi, aksioma OWL: `disjointWith`. Bu aksioma sinf misollari to'plamida umumiy elementlar mavjud emasligini aytishning bir usuli. Masalan, Man klassi Woman sinfi bilan umumiy elementlarga ega emasligini belgilashingiz mumkin. Shuningdek, sinflar bilan bog'liq bo'lgan qiymat cheklovlari ham mavjud bo'lib, ular rasmiy ravishda sinf aksiomalari emas, balki ularni tavsiflashning boshqa usulini ifodalaydi. Cheklovlar nomlar bilan ko'rsatilmagan, ammo bu

cheklovlarni qondiradigan misollar to'plami bo'lgan anonim sinflarni tavsiflaydi. OWLda bunday cheklovlarning ikki turi mavjud:

1. Qiymatlar bo'yicha cheklovlar (qiymat cheklovlari), agar biron bir aniq sinf sub'ekt sifatida harakat qilsa, berilgan qiymat ob'ektlariga o'rnatiladi. Bunday holda, ushbu xususiyatning ob'ektlarini atribut qiymatlari deb hisoblash tabiiydir. Qadriyatlariga bo'lgan cheklovlar qandaydir tarzda bunday qadriyatlar sifatida harakat qila oladigan shaxslar to'plamini cheklaydi. Siz, masalan, ma'lum bir xususiyatning barcha qiymatlari qandaydir aniq sinfning namunalari bo'lishi kerakligini aytishingiz mumkin. Ba'zi qiymat qiymatlariga global cheklov qo'yadigan rdfs: diapazondan farqli o'laroq, qiymatlar bo'yicha cheklovlar mahalliy bo'lib, ular faqat ma'lum bir sinf ularning predmeti bo'lgan taqdirdagina xususiyatlarga nisbatan qo'llaniladi. Misol uchun, FamilyCouple sinfi uchun siz Er va xotin xususiyatlariga cheklovlar qo'yishingiz mumkin, birinchi xususiyatda har doim Man sinfi, ikkinchisida esa - Ayol, agar faqat FamilyCouple klassi mavzu bo'lsa, ko'rsatishingiz mumkin.

2. Kardinallik cheklovlari (kardinallik cheklovlari), agar ma'lum bir sinf sub'ekt sifatida harakat qilsa, berilgan qiymat qiymatlari to'plamining asosiylikiga cheklovlar qo'yadi. Masalan, FootballTeam sinfining Player xususiyati uchun qiymatlar soni 11 bo'lishi kerakligini belgilashingiz mumkin.

Cheklovlar OWL yordamida tavsiflangan: Cheklov sinfi va OWL: onProperty xususiyati. OWL misoli: Cheklov klassi o'rnatiladi, odatda anonim, chunki u domenni tavsiflash uchun kerak emas va faqat yordamchi maqsadlarda ishlatiladi. Cheklovlar o'rnatilgan sinf OWLning ushbu anonim nusxasining ota-klassi sifatida e'lon qilinadi: Cheklov klassi. Masalan, FootballTeam klassi uchun cheklov uchun uni quyidagicha o'rnatishingiz mumkin:

Ushbu misolda biz OWLdan foydalandik: OWL-ga o'rnatilgan kardinallik cheklovi, agar ma'lum bir sinf mavzu bo'lsa, qiymat uchun qiymatlar sonini belgilash imkonini beradi. Sinf identifikatorini OWL da ma'lumotlar turi qiymatlari sifatida qo'llaniladigan odatiy qatordan ajratish uchun biz ikkita nuqta bilan FootballCommand konstruktsiyasidan foydalanamiz. The: Id konstruktsiyasi nomlar maydoniga ekvivalentdir: Id qurilishi, birinchi holatda nom maydoni sifatida faqat nomsiz nom maydoni ishlatiladi. Bunday nomlar maydonida e'lon qilingan identifikatorlar boshqa bo'shliqlardagi

identifikatorlardan ham farq qiladi, lekin bo'sh satr bo'shliq nomi sifatida ishlatiladi. Shunday qilib, yuqoridagi konstruktsiyada: FootballCommand klassi e'lon qilindi, bu OWLning misoli bo'lgan anonim sinfning pastki sinfi: Cheklov sinfi. Bunda anonim sinf owl: onProperty va owl: cardinality predikatlarining predmeti bo'lishi mumkin. Bu yerda OWL tiliga kiritilgan cheklovlar tavsifi. Bu cheklovlar yuqoridagi misolda ko'rsatilganidek, boshqa xususiyatlarning xususiyatlarini tavsiflovchi xususiyatlardir.

Qiymatlardagi cheklovlar:

OWL: allValuesFrom cheklovi, agar sinf ushbu xususiyatning predmeti bo'lsa, barcha xususiyat qiymatlari bir xil sinf misollarini ifodalashini ko'rsatish uchun ishlatiladi. Masalan, Person sinfi uchun hasParent xossasini kiritamiz va uning ushbu sinfdagi qiymatlari Person sinfiga tegishli ekanligini bildiramiz. OOP tillarida siz sinflar a'zolarini ularning turini ko'rsatgan holda belgilashingiz mumkin, bu cheklov bunday vazifaning analogidir. Person sinfida biz Person tipidagi hasParent xossasini o'rnatishimiz mumkin. Mantiqda OWLning analogi: allValuesFrom universallik miqdoriy ko'rsatkichidir.

OWL: someValuesFrom cheklovi mantiqdagi ekzistensial kvantning analogidir. Agar ushbu cheklov sinfning ba'zi bir mulkiga qo'yilgan bo'lsa, demak, bu sinf misollari bo'lishi mumkin - bu xususiyat bo'yicha mavzu sinfning ba'zi bir misollari bilan bog'langan qiymatlar. Misol uchun, siz Teacher sinfiga kirib, OWLni o'rnatishingiz mumkin: someValuesFrom cheklovi Person sinfning hasParent xususiyatiga ma'lum bir shaxsning ota-ona sifatida o'qituvchisi bo'lishi mumkinligini ko'rsatadi (albatta, ba'zi o'qituvchilarning farzandlari bor deb hisoblasak). Mantiqda ekzistensial miqdor ko'rsatkichi xuddi shu maqsadlarda qo'llaniladi.

OWL: hasValue cheklovi berilgan xususiyat kamida bitta ajratilgan qiymatga ega ekanligini ko'rsatish uchun ishlatiladi. Misol uchun, hasParent xususiyatiga Ivanov ismli shaxs qiymat bo'lishi kerakligini ko'rsatib, OWL: hasValue cheklovini o'rnatishingiz mumkin. Ushbu cheklov ota-onasi Ivanov ismli barcha shaxslar sinfini belgilaydi.

Quvvat chegaralari:

· OWL: maxCardinality cheklovi, agar tanlangan sinf uning mavzusi bo'lsa, ushbu xususiyat uchun maksimal qiymatlar sonini

belgilaydi. Masalan, aParent Person sinfiga ega bo'lgan xususiyatga cheklov o'rnatishingiz mumkin, bu xususiyat uchun qiymatlar soni ikkitadan ko'p bo'lmasligi kerak.

OWL: minKardinallik cheklovi ma'lum bir sinf sub'ekt sifatida ishlaydigan ma'lum bir qiymat uchun qiymatlarning minimal sonini belgilash uchun ishlatiladi. Oldingi misolda bo'lgani kabi, Person sinfining hasParent xususiyati uchun minimal qiymatlar soni ikkitaga o'rnatilishi mumkin.

OWL: kardinallik cheklovi ma'lum bir sinfnig ma'lum bir xususiyatining qiymatlari sonini aniq belgilashga imkon beradi. Masalan, siz OWL o'rnini bosishingiz mumkin: maxCardinality va OWL: min. Mulkning Ota-ona shaxsining bir OWL bilan chegaralanganligi: berilgan xususiyat uchun qiymatlar sonini ikkitaga o'rnatadigan asosiylik cheklovi.

Keling, har bir xususiyatni batafsil ko'rib chiqaylik. OWL: equivalentProperty. Bu xarakteristika ikkita xususiyatning bir xil juft shaxslar to'plamiga ega ekanligini ifodalash uchun ishlatiladi. OWL ekvivalentlik va xususiyatlarning tengligini ajratib turadi. Ekvivalentlik bu xususiyatlarning mazmunan mos kelishini anglatadi, ya'ni ular bog'laydigan shaxslar juftligi orqali. Tenglik OWL orqali ifodalanadi: bir xil qiymat kabi. Bu xususiyat faqat shaxslarni bog'laydi, shuning uchun u mulkka nisbatan qo'llanilsa, u holda bu xususiyatlar ham shaxslar sifatida ko'rib chiqilishi kerak. Bu faqat OWL Full-da mumkin, bu yerda sinflar va shaxslar o'rtasida hech qanday farq yo'q, ya'ni bu yerda sinfni o'ziga xos misol sifatida ko'rish mumkin. OWL: inverseOf. Ilgari ko'p marta ta'kidlanganidek, xususiyatlar ikkilik munosabatlardir, ya'ni ko'p juft shaxslar. Bu juftliklar tartiblangan, juftlikning birinchi elementi sub'ekt, ikkinchisi esa ob'ekt deb ataladi. Agar xususiyat boshqa xususiyatga teskari xususiyat sifatida tavsiflansa, bu xarakterlanadigan xususiyat bir xil juftliklar to'plamidan iborat bo'lishini anglatadi, lekin ularda sub'ekt va ob'ekt o'rnini o'zgartirgan. Misol uchun, agar sizda hasParent xossasi bo'lsa, unda siz hasChild konstruksiyasidan foydalanishingiz mumkin OWL: inverseOf hasParent, hasChild ning teskari xususiyatini yozishingiz mumkin. OWL: FunctionalProperty. Funksiyani birinchi elementlari ikkinchisini aniq belgilaydigan juftliklar to'plami (ikkilik munosabatlari) sifatida ham ko'rish mumkin. Ikkilik munosabat sifatida funktsiyaning muhim xarakteristikasi shundaki, agar individ

birinchi navbatda juftlikda paydo bo'lsa, u endi birinchi navbatda boshqa juftlikda paydo bo'lolmaydi. Aks holda, bu umumiy fikrga zid bo'ladi, chunki ta'rif sohasining har bir elementi domenning bitta elementi bilan mos kelishi kerak. Funktsional munosabatlarga misol sifatida hasHusband xossasini keltirish mumkin, u Woman sinfi misollarini Man sinfi misollariga moslashtiradi. Ayolning bir vaqtning o'zida bir nechta eriga ega bo'lishi mumkin emasligi taxmin qilinadi, shuning uchun bu mulkni funktsional qilish foydalidir. OWL: InverseFunctionalProperty. qiymat avvalgisiga o'xshaydi, faqat u birinchisiga emas, balki juftlik munosabatlarining ikkinchi elementlariga qaratilgan. Agar biron bir shaxs ikkinchi element sifatida biron bir juftlikda paydo bo'lgan bo'lsa, u endi munosabatlarning boshqa juftligida bu rolda paydo bo'lolmaydi. Masalan, ob'ektlari Ayollar sinfining misollari va qadriyatlari Erkaklar sinfining misollari bo'lgan Biologik Ona mulkidir. Bu xususiyatni qayta ishlash foydalidir, chunki har qanday odamda bir nechta biologik ona bo'lishi mumkin emas. OWL: TransitiveProperty. Tranzitivlik degani, agar berilgan munosabatda juftlik (x, y) va (y, z) bo'lsa, bu munosabatning juftlik to'plamida ham (x, z) bo'lishi kerak. Bu yerda (x, z) juftlik y elementi orqali "tranzit" hosil qiladi. O'tish munosabatlariga misol sifatida har qanday ierarxik munosabatlarni keltirish mumkin. Aytaylik, masalan, elementlari tekislikning hududlari bo'lgan Region sinfi mavjud. Ba'zi hududlar boshqalarda joylashgan bo'lib, ular subarea munosabati bilan ifodalanishi mumkin. Ko'rinib turibdiki, bu munosabat o'tishli, chunki agar hudud boshqa hududning pastki hududi bo'lsa, u ham ikkinchisini o'z ichiga olgan har qanday hududning pastki hududi hisoblanadi.

OWL: SymmetricProperty. Simmetrik munosabat quyidagi xususiyatni xarakterlaydi. Agar munosabatning juftlik to'plamida juftlik (x, y) bo'lsa, u holda juftlik (y, x) ham bo'lishi kerak. Shu ma'noda, munosabatlar faqat simmetrik juftliklardan iborat bo'lib, uning nomini belgilaydi. Nosimmetrik munosabatlarga misol sifatida Inson sinfining har bir misoli o'rtasidagi munosabatdir. Agar bir kishi boshqa do'st bo'lsa, buning aksi ham to'g'ri.

OWL dialektlari

OWL tilining uchta versiyasi mavjud: OWL Full. OWL to'liq dialekti, aslida, kengaytirilgan sintaksis RDF sxemasi tilidir. OWL

Full da OWL: Sinf sinfi rdfsdan farq qilmaydi: Class; OWLning boshqa dialektlarida har bir RDFS sinfi OWL sinfi emas. RDFSda bo'lgani kabi, OWL Full ham sinflar va shaxslarni ajratmaydi, ya'ni sinfni ham shaxslar yig'indisi sifatida, ham individual sifatida ko'rish mumkin. Misol uchun, OWL-da Full Tu-154 sinf nomi bo'lishi mumkin, ya'ni bir nechta samolyotni ifodalaydi, lekin AirplaneType sinfining namunasi ham bo'lishi mumkin. OWL Full barcha shaxslarni bitta qat'iy resurslar to'plamining a'zolari sifatida ko'radi (OWL: narsa rdfs ga teng: Resurs). Shuning uchun, ob'ekt xususiyatlari (OWL: ObjectProperty) va ma'lumotlar turi xususiyatlari (OWL: DatatypeProperty) o'rtasida hech qanday farq qilish mumkin emas. Tildagi elementlar o'rtasidagi munosabatlarga hech qanday maxsus cheklovlarning yo'qligi, umumiy holda, ushbu tilda amalga oshirilgan ontologiyalarga asoslangan xulosa chiqarish protsedurasi o'z vaqtida bajarilishini (yoki hatto tugashini) kafolatlashga imkon bermaydi. hammasi). OWL DL. Ushbu dialekt yuqorida muhokama qilingan Ta'riflovchi mantiqlar asosida ontologiyalarni tavsiflash imkoniyatini amalga oshiradi. Aniqrog'i, ushbu mantiqlardan biri SHIF (\mathcal{D}) deb ataladigan narsa amalga oshiriladi. Xulosa qilish tartibi bu yerda allaqachon hisoblab chiqiladi, ya'ni har doim o'z ishini maqbul vaqtda tugatadi. OWL DL turli cheklovlar bilan birga barcha OWL tili konstruktsiyalarini o'z ichiga oladi. Ulardan eng muhimlarini sanab o'tamiz:

OWL DL sinflar, ma'lumotlar turlari, ma'lumotlar turlarining xususiyatlari va ob'ekt xususiyatlari va shaxslarni ajratib turadi. Bu shuni anglatadiki, sinf bir vaqtning o'zida individual bo'la olmaydi.

- Yuqorida aytib o'tilganidek, OWL DL ob'ekt xususiyatlari va ma'lumotlar turi xususiyatlari o'rtasida farq qiladi. Bu shuni anglatadiki, OWL: inverseOf, OWL: InverseFunctionalProperty, OWL: TransitiveProperty va OWL: SymmetricProperty xususiyatlari ma'lumotlar turlarining xususiyatlariga qo'llanilmaydi.

- OWL DL o'tish xususiyatiga va ularning havolalariga asosiy cheklovlarni qo'llashga ruxsat bermaydi.

- OWL DL da RDF Schema lug'at elementlaridan foydalanishda sezilarli cheklov mavjud. Aslida, OWL DL RDFS ning deyarli barcha elementlaridan foydalanishni taqiqlaydi. OWL Lite. OWL Lite dialekti oddiy taksonomiyalarni loyihalash uchun mo'ljallangan bo'lib, unda siz qo'shimcha oddiy munosabatlarni ham e'lon qilishingiz

mumkin. OWL Lite SHIF (\mathcal{D}) mantiqini amalga oshiradi. OWL Lite OWL DL lahjasining barcha cheklovlarini saqlab qoladi, lekin o'ziga xos lahjani ham qo'shadi. Mana bu cheklovlardan ba'zilari:

- OWL Lite owl: hasValue, owl: disjointWith va boshqalar kabi cheklovlardan foydalanishni taqiqlaydi.
- OWL Lite OWL mavzusini talab qiladi: equivalentClass sinf nomi va bu predikatning ob'ekti sinf nomi yoki cheklovlar bilan hosil qilingan anonim sinf bo'lishi kerak. Xuddi shu narsa rdfs uchun talab qilinadi: subclassOf predikati.
- owl: allValuesFrom va owl: someValuesFrom predikat ob'ektlari sinflar yoki ma'lumotlar turlarining nomlari bo'lishi kerak.
- rdf predikatining ob'ektlari: tip sinf nomlari yoki cheklovlar bilan tuzilgan anonim sinf bo'lishi kerak.
- Predikat rdfs ob'ektlari: domen sinf nomlari bo'lishi kerak va rdfs predikat ob'ektlari: diapazon sinflar yoki ma'lumotlar turlarining nomlari bo'lishi kerak.
- Qiziqqan o'quvchi W3 konsortsiumi veb-saytida OWL tilining rasmiy tavsifida OWL dialektlari o'rtasidagi farqlarning batafsil tavsifini topishi mumkin.

OWL muharriri Protege

Protege, ehtimol, bugungi kunda mavjud bo'lgan eng mashhur OWL va RDF muharriridir. Muharrirni dastur veb-saytidan bepul yuklab olish mumkin. Dastur Mozilla Public License deb ataladigan dastur ostida tarqatiladi, bu sizga dasturdan hatto tijorat rivojlanishida ham bepul foydalanish imkonini beradi. Dastur Java tilida amalga oshiriladi va shuning uchun Java mashinasini oldindan o'rnatishni talab qiladi. Odatda, Java mashinasi Protege bilan birga foydalanuvchi kompyuteriga o'rnatilishining bir qismi sifatida tarqatiladi. Protege OWL ontologiyalarini yaratish uchun qulay muharrir bo'lib, biz uni aynan shu darajada ko'rib chiqamiz. Shunday qilib, keling, ushbu muharrirning imkoniyatlarini tavsiflashni boshlaylik. Bu yerda ko'rib chiqilayotgan dastur versiyasi 3.4 raqamiga ega va OWL tilining birinchi versiyasiga asoslangan. OWL 2 ni qo'llab-quvvatlaydigan Protege 4.0 ning beta-versiyasi ham mavjud.

Loyiha yaratish

Protege-dagi ontologiyalar loyihalar deb ataladigan shaklda yaratilgan. Loyiha - bu konfiguratsiya ma'lumotlari, birinchi navbatda,

tahrirlanadigan ontologiyani saqlaydigan OWL faylining nomi. Konfiguratsiya ma'lumotlari, shuningdek, ma'lum bir loyiha bilan bog'liq turli xil tahrirlash sozlamalarini saqlaydi. Tahrirlovchini ishga tushirganda, Protege foydalanuvchini mavjud loyihalarni tahrirlashga o'tishni yoki yangi loyiha yaratishni taklif qiladi.

Loyihani yaratish ontologiyani o'z ichiga olgan saqlash turini tanlash bilan boshlanadi. Biz "OWL / RDF fayllari" turini tanlaymiz. Keyingi tugmasini bosib va yaratilayotgan ontologiya nomini tanlash bosqichiga o'ting. OWL ontologiyalari Internetda noyob nomlarga ega bo'lishi kerak, shuning uchun ontologiya nomi veb-ma'lumotnoma hisoblanadi. Biz Protege muharriri sukut bo'yicha taklif qiladigan nomni tanlaymiz: <http://www.owl-ontologies.com/Ontology1237978552.owl>. Bunday holda, 1237978552 bizning ontologiyamizning muharrir tomonidan yaratilgan raqamli identifikatoridir. Keling, keyingi bosqichga o'tamiz. Bu yerda ontologiyamiz amalga oshiriladigan OWL lahjasini tanlash taklif etiladi. Shuningdek, siz RDF sxemasida ontologiyani amalga oshirishni tanlashingiz mumkin. Biz standart OWL DL dialektini tanlaymiz va loyihani yakunlash uchun Finish tugmasini bosib. Ontologiyani tahrirlash Loyihani yaratish tugallangandan so'ng dasturning asosiy oynasi ochiladi. Dastur oynasining yuqori qismida beshta yorliq mavjud bo'lib, ular funktsional jihatdan turli xil tahrirlash harakatlarini bir-biridan ajratishga imkon beradi. Keling, ushbu yorliqlarni sanab o'tamiz:

- **Metadata.** Ushbu yorliq sizga ontologiya haqidagi metama'lumotlarni tahrirlash imkonini beradi. Bular ontologiyaning o'zida tavsiflangan ma'lumotlarga bevosita aloqador bo'lmagan, balki ontologiyaning o'zini tavsiflovchi ma'lumotlardir. Bu yerda siz ontologiya nomini o'zgartirishingiz, ontologiya nomiga taxallus (prefiks) qo'shishingiz mumkin, bu ontologiyadan foydalanadigan boshqa dasturlar tomonidan ham, ontologiyaning o'zida ham qisqa nom sifatida ishlatilishi mumkin - prefiks deb ataladigan. Bu ontologiyada aniqlangan nomlarni boshqa ontologiyalarning nomlaridan ajratish imkonini beradi. Bundan tashqari, tahrirlanadigan ontologiyaga sharhlar yozishingiz mumkin.

- **OWL Classes** yorlig'i sizga yangi sinflar yaratish imkonini beradi. Siz yaratgan barcha sinflar OWLning pastki sinflari: Thing sinfi. Klasslar faqat nom bo'yicha emas, balki yuqorida tavsiflangan

mantiqiy yasovchilar, masalan, konyunksiya, diszyunksiya va boshqalar yordamida ham tuzilishi mumkin. Bundan tashqari, yuqoridagi cheklovlarni sinf xususiyatlarining qadriyatlarini va asosiyligiga qo'yishingiz mumkin.

- **"Xususiyatlar" yorlig'i** (Xususiyatlar) xususiyatlarni tahrirlash uchun turli funktsiyalarni o'z ichiga oladi. Ob'ekt xususiyatlari va ma'lumotlar turi xususiyatlarini yaratishingiz mumkin. Siz yaratayotgan mulkning domenini va diapazonini tanlashingiz mumkin, shuningdek, xususiyatning funktsionallik, teskari funktsionallik, simmetriya va o'tish qobiliyati kabi xususiyatlarini o'rnatishingiz mumkin. Teskari xususiyatning xarakteristikasi maxsus tarzda tanlanadi, chunki bu xususiyat teskari xususiyatga ega bo'lgan xususiyatni talab qiladi.

- **Individuallar yorlig'i** sizga sinflar misollarini aniqlash imkonini beradi. Yorliq ikki turdagi misollarni ko'rsatadi: foydalanuvchi tomonidan kiritilgan misollar (ta'kidlangan) va tizim tomonidan ontologiyada belgilangan qoidalarga asoslangan holda kiritilgan misollar (taxmin qilingan). Misol uchun, agar biror xususiyat boshqasiga teskari bo'lsa va bu xususiyat bilan bog'langan shaxslar juftligi kiritilgan bo'lsa, u holda tizim avtomatik ravishda teskari xususiyat uchun mos keladigan juftlikni kiritadi.

- **Shakllar yorlig'i** sizga sinflarning yangi namunalari kiritish uchun avtomatik ravishda shakllarni yaratishga imkon beradi. Shakl - maydonlari shu sinfning xususiyatlariga mos keladigan dialog oynasi. Maydonga kiritishingiz mumkin bo'lgan ma'lumotlar turlari ham qiymat qiymatlarining ma'lumotlar turlaridan meros bo'lib qoladi. Shubhasiz, faqat ma'lumotlar turlarining xossa qiymatlari shu tarzda o'rnatilishi mumkin.

Tahrirlovchi shuningdek, tahrirlangan ontologiya bo'yicha mantiqiy xulosa chiqarish va SPARQL ontologiyasiga so'rovlar o'rnatish imkonini beradi. Asosiy menyu bandini tanlaganingizda "Mulohaza yuritish" siz quyidagi amallarni bajarishingiz mumkin:

- Ontologiyaning mantiqiy izchilligini tekshiring (Check consentency).
- Ontologiyaning shaxslarini tasniflash uchun, ya'ni taksonomiya tuzing (Taksonomiyani tasniflash).
- E'lon qilingan sinf konstruktorlari yordamida yangi sinflarni hisoblang (Inferred tiplarni hisoblash).

- SPARQL tilida ontologiyaga so'rov o'rnatish.
- Yuqoridagi amallar bajariladigan mantiqiy xulosa dasturlarini o'zgartiring. 3.4 versiyasi ikkita shunday dasturni taklif qiladi: DIG Reasoner va Pellet 1.5.2.

Ta'riflangan operatsiyalarga qo'shimcha ravishda, Protege muharriri uslubiy sabablarga ko'ra bu yerda tavsiflanmaydigan yana ko'p narsalarni qilish imkonini beradi. Qiziqqan o'quvchi dastur veb-saytida joylashtirilgan muharrir bilan qanday ishlashni tavsiflovchi hujjatlarga murojaat qilishi mumkin.

Tavsifiy misol

Keling, qarindoshlik munosabatlarining eng oddiy ontologiyasini tuzamiz. Misol tariqasida Protege ontologiya kutubxonasi saytidan OWL fayli sifatida yuklab olish mumkin bo'lgan ta'lim maqsadlarida yaratilgan "family.swrl" ontologiyasini ko'rib chiqing. Ushbu misol, shuningdek, Semantik Web Rule Language (SWLR) imkoniyatlarini namoyish qilishni o'z ichiga oladi. SWLR tili OWL deb ataladigan ontologiya qoidalarini aniqlash uchun ishlatiladi, bu ontologiyaga yangi faktlar qo'shilganda bajariladigan harakatlarni qo'shish imkonini beradi. Misol uchun, siz ontologiyani amalga oshiradigan tizimga avtomatik ravishda juftlikni (x, y) qo'shishni aytadigan qoidani qo'shishingiz mumkin, agar faqat ba'zi bir x odamda ayol bola bo'lsa, Qiz munosabatlariga ega bo'lish uchun. Ushbu qoida quyidagicha ko'rinadi:

2.3. CYCL nima? CYCL uchun dasturlash tillari

Yuqorida aytib o'tilganidek, Cyc tizimining ontologiyasi ancha murakkab va tarmoqlangan tuzilishga ega. Ushbu bo'limda biz ushbu tuzilmani batafsilroq ko'rib chiqamiz. Cyc tizimining mualliflari Cyc ontologiyasining strukturasi 5.1-rasmda ko'rsatilgan piramida ko'rinishida ko'rsatishni afzal ko'radilar. Soddalashtirilgan tarzda, bu piramidani Cyc tizimi bilimlar bazasining sinflar (to'plamlar) taksonomiya daraxti tavsifi sifatida ko'rish mumkin. Albatta, Cyc tizimi sinfga bir nechta superklassga ega bo'lishga imkon beradi, shuning uchun aslida tizim ontologiyasining tuzilishi taksonomiya (daraxt) emas, balki yo'naltirilgan asiklik grafik, ya'ni yanada murakkab tuzilish. Taqdimotni soddalashtirish uchun biz ontologiyaning tuzilishini piramida shaklida tavsiflagan tizim

mualliflariga ergashib, strukturani taksonomiya sifatida ko'rib chiqamiz. Tizim mualliflari ushbu piramidani interaktiv qiladigan maxsus ilovani taqdim etishdi. Ilova sahifada mavjud. Foydalanuvchi faqat ontologiya piramidasiga figurasiga sichqonchanning chap tugmachasini bosishi kerak va interaktiv ilovaga ega yangi brauzer sahifasi ochiladi. Diqqatli o'quvchi, ehtimol, ontologiya piramidasida elementlarning nomlari bir-biridan sektorlar bo'yicha ajratilganligini payqagan bo'lishi mumkin. Agar foydalanuvchi sichqonchanning chap tugmasi bilan shunday sektorni bossa, piramida oynasida ontologiya elementi haqidagi ma'lumotlar ko'rsatiladi. Ilova oynasi uch qismga bo'lingan. Markaziy qismda ontologik piramidaning interaktiv chizmasi joylashgan. Chap qismda ontologiya elementining nomi ko'rsatiladi. Ushbu elementning tavsifi pastki qismida ko'rsatiladi. Ontologiyaning elementlari darajalarga bo'lingan, ularning har biri o'ziga xos rang bilan ta'kidlangan. O'ziga xoslikni qo'lga kiritganingizda, ya'ni taksonomiya daraxti orqali ko'tarilish, sathlarning rangi och kulrangdan to'q ranggacha o'zgaradi. Muayyan bilimlarning tavsifiga xos tushunchalar allaqachon quyuq ko'k rangda ta'kidlangan. Quyida biz piramidaning ko'p qismini batafsil ko'rib chiqamiz. Ontologiyaning ko'proq tanish ko'rinishi keltirilgan. Ushbu rasmda ontologiya yo'naltirilgan (rasmda qirralarning yo'nalishlari ko'rsatilmagan bo'lsa-da, har bir chekka yuqori sinfdan bolalar sinfigacha bo'lgan yo'nalishga ega deb taxmin qilinadi) asiklik grafik sifatida tavsiflangan.

Och kulrang daraja

Bu daraja hali fazoviy va vaqtinchalik miqyosga ega bo'lmagan eng mavhum tushunchalarni o'z ichiga oladi. Keling, ularni ko'rib chiqaylik. Thing Cyc tizimi taksonomiya daraxtining ildizidir. Ontologiyaning har bir elementi yuqori sinf sifatida Ob'ekt sinfiga ega. Demak, Cyc sistemasi ontologiyasining har bir elementi Ob'ekt deb aytishimiz mumkin. Amalga oshirish nuqtai nazaridan, Ob'ekt tushunchasi sizga quyidagilarga imkon beradigan ma'lumotlarni o'z ichiga olgan tuzilmadir:

- 1. Taksonomiya moddalarini bir-biri bilan solishtiring.*
- 2. Dasturlash interfeysi yordamida ontologiya elementlariga bevosita kirish.*

Nomoddiy narsa sinfi Ob'ekt sinfining bevosita avlodidir. Nomoddiy ob'ektlar moddiy timsolga ega emas, ular yasalmagan,

kodlanmagan va hokazo. Nomoddiy narsalar tog'larda sayr qilish yoki nonushta qilish kabi hodisalardir. G'oyalar ham nomoddiy ob'ektlardir. Amalga oshirish nuqtai nazaridan, nomoddiy narsalarni boshqalardan ajratish juda foydali. Bu turli xil xatolardan qochadi, masalan, foydalanuvchidan keyingi payshanba kuni uchrashuv qanday rangda bo'lishini so'rash. Ob'ekt sinfining yana bir bevosita avlodi Individual sinfdir. Individuallarda asosiysi ular to'plamlar va kolleksiyalardan farq qiladi. Individlar hodisalar, jismoniy ob'ektlar, raqamlar va munosabatlardir. O'ziga xos Toyota Corolla avtomobili - bu Individual, ammo "to'rt eshikli sedan" kontseptsiyasi allaqachon to'plamdir. Ushbu to'plam, beton mashinadan farqli o'laroq, mavhumdir. U kengaytma, massa va maksimal tezlikka ega emas, lekin elementlar, subkolleksiya va superkolleksiyalarga ega. Relation sinfi nomoddiy ob'ekt sinfining avlodidir. Aloqalar bir-biridan farq qiluvchi tushunchalarni hukmlarga bog'laydigan elimga o'xshaydi. Masalan, "Jozef Kobzon" va "qo'shiqchi" tushunchalarini "Jozef Kobzon - qo'shiqchi" jumlasini bilan bog'lash mumkin. Aloqador tushunchalar hukmlardir. Bundan tashqari, aloqalardan kompaniya ma'lumotlar bazalari yoki veb-sahifalar kabi tashqi bilim manbalari bilan ishlashda foydalanish mumkin. O'quvchi yuqorida ko'rib turganidek, Cyc tizimi printsip asosida qurilgan: "qanchalik ko'p hukmlar bo'lsa, shuncha yaxshi". Tizim yaratuvchisi Duglas Lenat o'zining "Aql - bu o'n million hukm" degan gapini bilan mashhur. CycL birinchi tartibli predikatlar hisobi tilidir, lekin u ikkinchi tartibli tilga tegishli ko'plab xususiyatlarni ham o'z ichiga oladi. Xususan, munosabatlar boshqa munosabatlar orqali ham bir-biri bilan bog'lanishi mumkin, bu boshqa narsalar qatorida tizimni avtomatik ravishda o'rganish imkonini beradi. Tizimga Car X havo yostig'i qo'shilsa, Cyc tizimi allaqachon Car X da havo yostig'i va uning barcha qismlari avtomobil ichida mavjudligini biladi. Bu Contains, Being Part va Within munosabatlari o'rtasidagi mavjud aloqalar natijasida sodir bo'ladi.

Set klassi matematik to'plam tushunchasining bevosita analogidir. To'plamdan farqli o'laroq, to'plam elementlari bir-biri bilan umumiy xususiyatlar bilan bog'liq bo'lishi shart emas. Bunday elementlarning yagona umumiy xususiyati shundaki, ular berilgan to'plamga tegishlidir. Collection klassi biz birinchi bobda muhokama qilgan tip tushunchasining bevosita analogidir. To'plamning barcha

elementlari umumiy narsaga ega va bu umumiylik bir xil atributlar orqali ifodalanadi. Yuqorida aytib o'tilganidek, to'plamlar va shaxslardan to'plamlar o'rtasidagi asosiy farq shundaki, ikkinchisida elementlar bo'lishi mumkin emas. To'plamlar to'plamlardan ham farq qiladi, chunki ular bir vaqtning o'zida bir nechta "mujassamlanish" ga ega bo'lishi mumkin, ya'ni Bir xil elementlarga ega Collection sinfining ikkita nusxasi bo'lishi mumkin, ammo ular baribir boshqacha ko'rib chiqiladi. Agar ular bir xil elementlarni o'z ichiga olsa, to'plamlar bir xil deb hisoblanadi. Umuman olganda, yaxshi to'plam - bu elementlarning oddiy ro'yxati bilan aniqlash qiyin yoki imkonsizdir. Misol uchun, barcha oq avtomobillar to'plami yaratish uchun juda yaxshi namuna emas, chunki bu kontseptsiya Cyc tizimining boshqa elementlari yordamida osongina aniqlanadi va shuning uchun kontseptsiyaning o'zi unchalik ma'lumotga ega emas. Boshqa tomondan, barcha oq yoqalarning to'plamini oddiygina elementlarni sanab o'tish orqali aniqlash juda qiyin, shuning uchun bu kontseptsiya haqida ko'p gapirish mumkin. Path klassi kesishmaydigan yo'l yoki halqadir. Yo'llar fazoviy-vaqtinchalik abstraktsiyalarni (grafiklardagi tugunlar orasidagi qirralar), joylashuvning nomoddiy abstraktsiyalarini (kenglik chiziqlari) yoki juda aniq chiziqlarni (magistral yo'lidagi ajratuvchi chiziq) ifodalashi mumkin. Boshqacha qilib aytganda, yo'l ma'lum bir tizimda yo'l sifatida ishlatiladigan har qanday narsa bo'lishi mumkin. Misol uchun, apelsin odatda boradigan yo'l emas. Ammo bu apelsinlar yordamida chumoli yo'lini boshqarish tizimida chumoli uchun yo'lga aylanishi mumkin. Shuni ham yodda tutingki, Yo'l individual shaxs emas, chunki Hech narsa ma'lum tizimlarda to'plamlar yoki to'plamlarni yo'l sifatida ishlatishingizga to'sqinlik qilmaydi. Yo'llarni asosiy ajratish ochiq egri chiziqlarni ifodalovchi Simple Path va barcha yopiq yo'llarni ifodalovchi Cyclic Path deb ataladi. Mantiq tushunchasi Cyc ontologiyasida to'plam bo'lmasa-da, bu tushuncha unda asosiy hisoblanadi. Hukmlar deb ataladigan Cyc bilimlar bazasining elementlari mantiqiy xulosalar asosida qurilgan. Shuning uchun bo'lsa kerak, mualliflar mantiq bilan bog'liq tushunchalarning tavsifini ochiq kulrang darajaga qo'yishadi. Biz bu tushunchalarni yuqorida batafsil muhokama qildik, shuning uchun bu yerda tavsifni takrorlashning hojati yo'q. Qisqacha aytaylik: Cyc ikkita mantiqiy konstantaga ega True va False. Hukmlar konstantalar va boshqa hukmlardan mantiqiy

bog'lovchilar - konyunksiya, disjunksiya, inkor va implikasiya operatorlari yordamida quriladi. Agar hukm uning konstantalarining ba'zi bir to'plami bo'yicha bajarilgan bo'lsa, unda bu hukm haqiqat deb hisoblanadi, aks holda u noto'g'ri hisoblanadi. Bundan tashqari, Cyc yuqorida tavsiflangan xulosa chiqarish tartibini ham amalga oshiradi. Quyi darajalar (fazoviy, vaqtinchalik yoki moddiy) tushunchalari bilan bog'liq bo'lmagan matematik (yoki hisoblash) narsalar ham shu darajada joylashgan. Bunday tushunchalar qatoriga raqamlar, to'plamlar, algoritmlar, mavhum belgilar qatorlari va boshqalar kiradi.

O'rtacha kulrang daraja

Bu darajada fazoviy va vaqt munosabatlari bilan bog'liq bo'lgan, shuningdek, materiya tushunchasi bilan bog'liq bo'lgan tushunchalar mavjud. Ba'zi tushunchalar Cyc ontologiyasida sinflar (to'plamlar) shaklida ko'rsatilgan, boshqalari o'xshash xususiyatlar bilan birlashtirilgan tushunchalarning butun guruhlarini. Keling, ularni ko'rib chiqaylik. Spatial Thing sinfi kenglik va joylashuv bilan bog'liq barcha tushunchalarni o'z ichiga oladi. Ushbu sinf tushunchalari bilan tanishishda siz ikkita narsaga e'tibor berishingiz kerak. Birinchidan, Space Class sinfining ob'ekti qisman moddiy (masalan, Leningrad viloyati) yoki barcha nomoddiy (tekislikdagi doira) bo'lishi mumkin. Ikkinchidan, buni amalga oshirish mumkin bo'lsa-da, boshqa xususiyatlar bilan bog'liq holda, ushbu ob'ektning joylashishini aniqlash mutlaqo kerak emas. Gap shundaki, har bir xususiyatning joylashuvi mavjud emas. Masalan, tekislikdagi segment geometrik tushuncha sifatida joylashuvga ega emas. Spatial Path klassi xususiyatlarni bog'laydigan kosmosdagi yo'llarni ifodalaydi. Fazoviy yo'llarga misollar qatoriga yo'llar, koridorlar, simlar, qon tomirlari va nervlar kiradi. Shuni yodda tutish kerakki, oila daraxti kabi mavhum yo'llar Kosmik yo'llar emas. Border klassi ikkita Xususiyatlar orasidagi chegaralar (shakl ramkalari) bo'lib xizmat qiluvchi chiziqlar, chiziqlar, tekisliklar yoki fazoviy hududlar kabi tushunchalarni o'z ichiga oladi. Harakat sinfi fazoviy harakat hodisasidir. Ko'chirish ma'lum bir yo'l bo'ylab sodir bo'lishi mumkin, ya'ni ob'ekt, bu holda, ma'lum masofani bosib o'tadi yoki ob'ektning ba'zi o'q bo'ylab aylanishi bo'ladi. Harakat doimiy yoki davriy bo'lishi mumkin. Harakatga misol tariqasida samolyotda uchish (to'xtab turish), ovqat iste'mol qilish yoki elektr toki urishi kiradi. Artifakt sinfi agentlar tomonidan maxsus maqsadda ishlab chiqarilgan jonsiz narsalarni

ifodalaydi. Artefakt yaratish uchun uning uchun yangi material yaratish shart emas, agent shunchaki boshqa narsalarni qayta ishlash orqali artefakt yaratishi mumkin. Artefaktlarga misol qilib, daraxt shoxlaridan yasalgan nay yoki metallardan bo'rtirma yo'li bilan yasalgan tangalar kiradi. Bularga inson ashyolari (asboblar, binolar, hujjatlar, elektr uzatish liniyalari) bilan bir qatorda, hayvonlar yasagan artefaktlar (uyalar, termit tepaliklari, qunduz to'g'onlari) kiradi. Cyc tizimi 37 mingdan ortiq turli xil hodisalardan foydalanadi. Voqealar dunyoda sodir bo'lgan voqealarni tavsiflash uchun ishlatiladi. Umumiy hodisalarga qo'shiq aytish, xarid qilish, fikrlash ("fikrlash" hodisasi sifatida qaraladi) va boshqalar kiradi. Skript (skript) - vaqtinchalik munosabatlar bilan o'zaro bog'langan kichik hodisalardan tashkil topgan murakkab hodisaning tavsifi. Ilovalar ma'lum bir voqea ma'lum skript tomonidan tavsiflangan murakkabroq hodisaning bir qismi ekanligini tan olish uchun skriptlardan foydalanishi mumkin. Skriptlardan o'qish jarayonini rejalashtirish va amalga oshirish uchun ham foydalanish mumkin.

Temporal Thing klassi vaqt oralig'i yoki joylashuviga ega bo'lgan ob'ektlarni ifodalaydi. "Qachon?" Savoli ushbu turdagi ob'ektlarga tegishli. Shunday qilib, ko'p narsalarni vaqtinchalik ob'ektlar sifatida tasniflash mumkin, masalan, hodisalar, jismoniy ob'ektlar, kelishuvlar, vaqt oraliqlari va boshqalar. Matematik to'plamlar, atributlar yoki raqamlar kabi vaqtinchalik xususiyatiga ega bo'lmagan ob'ektlar vaqtinchalik ob'ektlar emasligi aniq.

To'q kulrang daraja

Ushbu darajada ob'ektlar moddiy timsolga ega yoki bunday ob'ektlar bilan o'zaro ta'sir qiladigan joylashadi. Keling, ushbu darajadagi elementlarni batafsil ko'rib chiqaylik. Physical Object sinfi materiyadan tashkil topgan va vaqt ichida mavjud bo'lgan barcha ob'ektlarni ifodalaydi (ya'ni, bunday ob'ektlarga vaqt toifalari qo'llanilishi mumkin: qachon paydo bo'lgan, qanday o'zgargan va hokazo). Bunday ob'ektlar nomoddiy elementlarni o'z ichiga olishi yoki bo'lmasligi mumkin. Masalan, monografiyaning o'ziga xos nusxasi jismoniy ob'ektdir, chunki materiyadan tuzilgan va fazoviy va vaqtinchalik hajmga ega (hajmi va nashr etilgan yili). Lekin, bu ob'ekt ham mazmunga ega, ya'ni ushbu monografiya sahifalarida chop etilgan matnning ma'nosi. Tarkib, shubhasiz, nomoddiy. Materiallar jonsiz ob'ektdan birinchi ob'ektni yaratish uchun ishlatilgan boshqa

jonsiz ob'ektni tanlash uchun ishlatiladi. Materialni qisman boshqa ob'ektdan tashkil topgan alohida ob'ekt va bu ob'ekt o'rtasidagi bog'liqlik sifatida ko'rish mumkin, bu esa o'z navbatida birinchi ob'ektda ko'proq yoki kamroq teng taqsimlanadi. Misol uchun, bir stakan choyda erigan qoshiq shakar bu ichimlik uchun material bo'ladi. Qismlar shaxslar va ularning tarkibiy qismlari o'rtasidagi bog'lanish sifatida qaraladi. Komponentlar bu yerda keng kontekstda tushuniladi. Bu, masalan, fazoviy komponentlar, vaqtinchalik komponentlar, kontseptual komponentlar, guruh a'zolari va boshqalar bo'lishi mumkin. Bo'laklar predmetning ma'lum ma'noda butunning bir qismi ekanligini aytish uchun ishlatiladi. Cyc ontologiyasida eng ko'p ishlatiladigan qismlar jismoniy elementlar, kichik hodisalar, vaqt oralig'i va guruh a'zolaridir. Static Situation klassi Temporal Situation sinfining kichik sinfidir. Har bir statik holat - bu ma'lum vaqt oralig'ida statik ravishda mavjud bo'lgan ikki yoki undan ortiq ob'ektlar o'rtasidagi qandaydir bog'lanish holati. Statik vaziyatlar har doim vaqtinchalik o'lchovga ega va ko'pincha fazoviy o'lchov va moddiy timsolga ega. Jismoniy holatning o'zgarishi va o'zgartiruvchi hodisalar (Metamorfoz hodisalari) va shunga o'xshash boshqa hodisalar nimadir mavjud bo'lishni to'xtatib, boshqa narsa boshlanishida ifodalangan o'zgarishlar sifatida qaralishi mumkin. Physical Agent klassi mavjud va hech bo'lmaganda moddiy timsoliga ega bo'lgan ob'ektlarni ifodalaydi. Har bir Jismoniy Agentning xohish va niyatlari, shuningdek, bu istak va niyatlarni amalga oshirish qobiliyati bor. Agentlar individual bo'lishi mumkin yoki birgalikda harakat qiluvchi bir nechta agentlardan iborat bo'lishi mumkin.

Reja - bu to'g'ri vaqt tartibida bajarilganda, biron bir maqsadga erishish yoki biron bir vazifani amalga oshirishga olib keladigan bosqichlar ketma-ketligi. Maqsadlar Cyc tizimida maqsadlar agentlar va narsalarning holatini tavsiflovchi formulalar o'rtasidagi munosabatlarning xususiyatlari bo'lib, ularga erishish uchun ushbu agentlar ma'lum qadamlar qo'ymoqchi. Shuningdek, maqsadni agent va ba'zi Cyc formulasi bilan tavsiflangan statik vaziyat o'rtasidagi munosabatlarning atributi sifatida ko'rish mumkin. Aktyorlar hodisa va bu hodisaga qandaydir tarzda jalb qilingan har qanday mavjud ob'ekt o'rtasidagi munosabatlarning bir turi. Boshqacha aytganda, biror kimsa yoki biror narsa muayyan hodisaning aktyori (aktyori) deyilganda, uning borishi davomida bu kimdir yoki nimadir qandaydir

tarzda ishtirok etganligi tushuniladi. Shu bilan birga, aktyor har doim ham tadbirda faol rol o'ynamasligini ta'kidlash kerak. Misol tariqasida "Chexov miltig'i"ni keltirish mumkin, agar u harakat paytida devorga osilib qolsa, spektakl oxirida o'q uzilishi kerak. Harakatlar - bu qandaydir aktyor tomonidan amalga oshiriladigan voqealar. Harakatlar - bu bir yoki bir nechta aktyorlar dunyoning holatini biroz o'zgartiradigan, odatda ma'lum bir kuch sarflashda ifodalangan voqealar. Shu bilan birga, buning natijasida moddiy o'zgarishlar (ob'ektlarning ko'chishi yoki ularni yo'q qilish) sodir bo'lishi shart emas. Masalan, mijozning bank hisobini o'zgartirish bo'lishi mumkin. Bundan tashqari, aktyor jonli ob'ekt bo'lishi shart emas. Misol uchun, 2009 yilda Moskvaning Ozernaya ko'chasidagi gaz quvuridagi avariya, o'shanda ko'plab avtomobillar yonayotgan gaz yong'inidan zarar ko'rgan (bu yerda aktyor gaz mash'alasi). Agent klassi istak va niyatlarga ega bo'lgan ob'ektlarni, shuningdek, ushbu istak va niyatlarni amalga oshirish qobiliyatini ifodalaydi. Agentlar individual bo'lishi mumkin yoki birgalikda harakat qiluvchi bir nechta agentlardan iborat bo'lishi mumkin.

Binafsha daraja

Bu daraja inson odatda uning atrofidagi dunyo haqidagi bilimi deb hisoblaydigan bilimlarni o'z ichiga oladi, ya'ni uning turli jihatlari haqida. Bunday bilimlar qatoriga quyosh tizimi, uning tuzilishi, sayyoralar soni va boshqalar haqidagi bilimlar kiradi. Chuqur fazo haqida bilim bor, ya'ni yulduzlar, galaktikalar va boshqa shunga o'xshash ob'ektlar haqida, bu bilimlar "astronomiya" bo'limida birlashtirilgan. Inson atrofidagi ob'ektlar haqida tasavvurga ega, shuning uchun Cyc-da bizning dunyomizning ko'plab ob'ektlariga ega bo'lgan maxsus jismoniy artefakt sinfi mavjud. Moddalarning tuzilishi haqidagi bilimlar "Kimyo" bo'limiga birlashtirilgan. Ushbu bo'limda "Analitik kimyo", "Hisoblash kimyosi", "Elektrokimyo" va hokazo turdagi kichik bo'limlar mavjud. Materiklar, dengizlar va okeanlar haqidagi bilimlar "Tabiiy geografiya" deb nomlangan bo'limga, dunyoning geosiyosiy tuzilishi haqidagi bilimlar esa "Siyosiy geografiya" bo'limiga birlashtirilgan. Dunyo siyosiy xaritasining elementlari - bu davlatlar, hukumatlar va geosiyosiy ob'ektlar (muayyan mintaqani boshqaradigan odamlar guruhlar). Shuningdek, har xil turdagi tashkilotlar, ularning turlari, shuningdek, tijorat va harbiy tashkilotlar haqida ma'lumot mavjud. Cyc tizimi, shuningdek,

umuman tirik organizmlar (Tirik mavjudotlar), shuningdek, o'ziga xos o'simliklar va hayvonlar haqida biladi. Ekologiya, ijtimoiy xulq-atvor va huquq bo'yicha bilimlar mavjud.

Moviy, o'rta ko'k va to'q ko'k darajalari

Moviy darajada, binafsha rangda bo'lgani kabi, atrofdagi dunyoning turli tomonlari haqida ma'lumot mavjud, ammo allaqachon aniqroq. Misol uchun, bo'limlardan biri avtomobillar, binolar va qurollarning tavsifiga bag'ishlangan, ya'ni inson hayotining o'ziga xos tomonlari haqidagi bilimlarning tavsifini o'z ichiga oladi. Yana bir shunga o'xshash bo'lim sanoat mahsulotlari va turli mexanizmlar haqidagi bilimdir. Bu daraja tillar, adabiyot va san'atning boshqa sohalarini bilishni ham o'z ichiga oladi. Moviy daraja, shuningdek, ijtimoiy faoliyatning turli sohalariga oid bilimlarni o'z ichiga oladi. Bu jamiyat tuzilishi haqidagi bilimlar, shuningdek, ushbu tuzilmaning elementlari to'g'risidagi aniq bilimlardir. Masalan, sport, savdo va savdo, sayohat, kasblar, siyosat va urushlar va boshqalar haqida bilimlar mavjud. O'rtacha ko'k darajada inson faoliyatining muayyan sohalarini haqidagi bilimdir. Ushbu bilimlar sog'liqni saqlash, kompyuter xavfsizligi, kimyoviy reaksiyalarning ayrim turlarini bilish va boshqalarni o'z ichiga oladi. To'q ko'k daraja aniq faktlar va bilimlarni o'z ichiga oladi. Bu odatda ma'lumotlar bazasi jadvallarida qayd etilgan bilimdir. Biroq, ma'lumotlar bazalaridan farqli o'laroq, Cyc-dagi maxsus bilimlar yuqori darajadagi tushunchalar bilan bog'liq. Masalan, Cyc tizimi bir necha yuz mashhur musiqachilarning bilimlarini o'z ichiga oladi. Cyc tizimi musiqachilar biladigan 85 ga yaqin narsani biladi. Shuning uchun, masalan, "Vladimir Spivakov" shunchaki jadvaldagi yozuv emas, Cyc Spivakov skripkada o'ynashini, "Moskva virtuozi" ansamblini boshqarishini va hokazolarni biladi. Umuman olganda, Cyc tizimi mavjud ma'lumotlar bazalarini tizimga birlashtirish imkoniyatiga ega. Jadval ustunlarini Cyc ontologiyasi tushunchalari bilan aniqlab, ma'lumotlar bazasini Cyc tizimining bilimlar bazasiga joylashtirish mumkin. Keyinchalik, jadvallarning yozuvlari Cyc tizimi tomonidan hukm sifatida talqin qilinadi va ma'lumotlar bazasiga savollar berish, ushbu ma'lumotlar bazalarida joylashgan faktlar asosida xulosalar chiqarish va h.k.

OpenCyc loyihasi

Yuqorida aytib o'tilganidek, OpenCyc loyihasi Cyc tizimining ochiq manba versiyasidir. OpenCyc quyidagi uchta elementdan iborat:

- Xulosa qilish mexanizmi.
- OpenCyc bilimlar bazasi brauzeri.
- OpenCyc ontologiyasi bilan o'zaro aloqada bo'lishi mumkin bo'lgan dasturlarni yozish uchun interfeys funksiyalari to'plami (API).

Yuqorida biz Cyc inference mexanizmi qanday ishlashini batafsil muhokama qildik va interfeys funksiyalarining tavsifi asosan dasturchilar uchun qiziqarli. Shuning uchun, bu yerda biz OpenCyc ma'lumotlar bazasi brauzeriga e'tibor qaratamiz. OpenCyc loyihasining sayti ham qiziqarli, chunki u juda batafsil hujjatlarga ega. Nashr etilgan hujjatlar orasida siz OpenCyc bilimlar bazasi bilan ishlashning turli jihatlari bo'yicha bir qator o'quv qo'llanmalarini topishingiz mumkin. Shuningdek, CycL tilida ontologiyalarni yozish uslublari va ontologiyalarni yaratish bilan bog'liq turli jihatlar batafsil tavsiflangan. Masalan, hodisalar va ular bilan bog'liq tushunchalar diqqat bilan tavsiflangan. Muallif qiziqqan o'quvchiga loyiha saytining mazmunini ko'rib chiqishni tavsiya qiladi, albatta, qiziqarli narsa topiladi. Ushbu maqola yozilayotganda, OpenCyc ishlab chiquvchilari tizimning ikkinchi versiyasini chiqarmoqchi edilar, ammo u ustida ishlash hali tugamagan edi. Shu munosabat bilan, bu yerda tizimning birinchi versiyasining tavsifi - Cyc-101 deb ataladi. To'g'ri aytganda, bu yerda faqat tizim interfeysi haqida umumiy ma'lumot berilgan. Muallif batafsil tavsif berishni kerak emas deb hisobladi, chunki loyiha saytida tizim bilan barcha ishlar yaxshi hujjatlashtirilgan. Quyida tizimning imkoniyatlarini ko'rsatishga qaratilgan kichik bir ko'rinish berilgan, boshqa hech narsa yo'q.

Bilimlar bazasi brauzeri

OpenCyc ontologiya brauzeri foydalanuvchi kompyuterida mahalliy o'rnatilgan veb-server bo'lib, ushbu server tomonidan taqdim etilgan interfeyslar orqali OpenCyc ontologiyasiga kirish imkoniyatini beradi. Serverni o'rnatish bo'yicha batafsil ko'rsatmalar loyiha veb-saytida berilgan, shuning uchun biz bu yerda bunday narsalarga e'tibor bermaymiz.

1. Asboblari paneli (Tools Frame yoki Toolbar Frame) foydalanuvchiga quyidagi funksiyalarni beradi:

a) Bilimlar bazasidan konstantalarni "To'ldirish" maydoniga kiritish orqali qidiring. Qidiruv uchun uchta tugmadan foydalanish mumkin: "To'liq", "Show" va "Grep". To'liq atamasi To'ldiruvchini anglatadi, ya'ni agar foydalanuvchi maydonga doimiy nomning

boshini kiritisa, u holda qidiruv natijalari kiritilgan qatordan boshlanadigan barcha nomlarni ko'rsatadi. "Ko'rsatish" tugmasi nomlari maydonga kiritilgan nomga to'liq mos keladigan konstantalarni ko'rsatish uchun ishlatiladi (odatda, bunday nomlar ingliz tilida kiritiladi va doimiylarga sharhlarda qidiriladi. Masalan, foydalanuvchi qatorni kiritishi mumkin " Uilyam Jefferson Klinton" va natijada doimiy \$ # BillKlinton indeks panelida ko'rsatiladi). Va nihoyat, "Grep" tugmasi mashhur UNIX grep qidiruv yordam dasturida bo'lgani kabi, qidiruvda muntazam ifodalardan foydalanishga imkon beradi. Shu bilan bir qatorda, Clear tugmasi kiritish maydonini tozalash uchun ishlatilishi mumkin.

b) "Asboblar" va "Navigatsiya" giperhavolalari yordamida qo'shimcha funktsiyalarni ta'minlaydigan brauzer sahifalariga o'ting.

c) Boshqa foydalanuvchi nomi bilan tizimga kiring. Buni "Login" giperhavolasi orqali amalga oshirish mumkin.

2. Index Frame Cyc ontologiyasida tanlangan konstantalar haqidagi ma'lumotlarni ko'rsatish uchun ishlatiladi. Panelda panelda ko'rsatilgan doimiylar ustida bajarilishi mumkin bo'lgan operatsiyalar ro'yxati, shuningdek, ushbu konstantalarni o'z ichiga olgan hukmlar turlari ro'yxati mavjud. Amaliyotlar ro'yxatining mazmuni indeks panelining mazmuniga bog'liq, ya'ni hozirda ushbu panelda ko'rsatilgan doimiylardan. Masalan, panelda ko'rsatilgan konstantalar bilan ishlatiladigan evristik xulosa modullariga havolalar ham bu yerda ko'rsatilishi mumkin.

3. Assertion Display Frame indeks panelida tanlangan konstantalar bilan bog'liq bo'lgan mulohazalarni ko'rsatish uchun ishlatiladi. Shubhasiz, hukm panelining mazmuni indeks panelining mazmuniga bog'liq. Har bir hukm uchun ushbu hukm tegishli mikroteoriya ko'rsatilgan. Foydalanuvchi ushbu qaror uchun operatsiyalarni tahrirlash uchun qaror giperhavolasini kuzatishi mumkin. Tahrirlash operatsiyalari sizga hukmni o'zgartirish, uni o'chirish yoki shunga o'xshash hukmni bilimlar bazasiga qo'shish imkonini beradi. Har bir hukm ma'lum bir rangdagi to'p bilan belgilanadi. To'pning rangi berilgan hukmning haqiqat qiymatini ifodalaydi. Biz yuqorida hukmlarning haqiqat qadriyatlarini muhokama qildik. Oq to'p hukm har doim to'g'ri ekanligini anglatadi va sariq to'p sukut bo'yicha hukm haqiqat ekanligini anglatadi.

4. Systems Information Frame foydalanuvchiga Cyc tizimining joriy holati haqida ma'lumot beradi. Ushbu ma'lumotni Yangilash giperhavalasini bosish orqali yangilash mumkin. Tizim ma'lumotlari, asosan, tizimda hozirda amalga oshirilayotgan operatsiyalar to'g'risidagi ma'lumotlarni o'z ichiga oladi. OpenCyc ma'lumotlar bazasi brauzeri tomonidan taqdim etilgan ba'zi qiziqarli vositalar quyida muhokama qilinadi.

ASK operatsiyasi

Ma'lumot bazasi brauzeri Cyc tizimiga savollar berish imkonini beradi, ya'ni xulosa chiqarish operatsiyalarini bajarish. Buning uchun asboblardan panelidagi "So'rash" giperhavalasini bosishingiz kerak. Natijada, OpenCyc ontologiyasi uchun so'rov sahifasi ko'rsatiladi. Foydalanuvchi so'raladigan mikro nazariyani tanlashi mumkin, buning uchun "To'ldirish" tugmasidan ham foydalanish mumkin. Bundan tashqari, so'rovdagi konstantalarni "\$ #" bosh belgisisiz kiritish mumkin, ammo bu holda siz "Cyclify" tugmasini bosishingiz kerak (ularni CycL tiliga moslashtiring) va nomlar bilan birga qo'shiladi. "\$ #" prefiksi. So'rovning o'zi "So'rov so'rash" tugmasini bosish orqali bajarilishi mumkin. Naqshlarga asoslangan hukmlarni qo'shish yana bir qiziqarli vosita - shablon sifatida OpenCyc bilim bazasida mavjud bo'lgan ba'zi mulohazalar yordamida OpenCyc ontologiyasiga yangi hukmlarni qo'shish qobiliyati.

XULOSA

Bu monografiya muhandislik muammolarini hal qilishda ontologiyalardan foydalanishning turli jihatlarini qamrab olgan. O'quvchini mashina bilimlarini qayta ishlashda yuzaga keladigan muammolar qatori bilan tanishtirish, ushbu muammolarda ontologiyalarning rolini va ikkinchisini maxsus o'rganish zarurligini ko'rsatish uchun zarur bo'lgan birinchi bob bundan mustasno. Monografiyaning qolgan qismi, aslida, u ontologiya kutubxonalarini qurish bo'yicha loyihalarni yozish paytidagi eng mashhurlarining umumiy ko'rinishidir. O'quvchi matnni o'qish jarayonida qayta-qayta ishonch hosil qilish imkoniyatiga ega bo'lganligi sababli, bilimlarni qayta ishlashning yaxshi loyihasini yaratishni ushbu bilimlarning ontologiyasi kutubxonasini taqdim etmasdan turib tasavvur qilib bo'lmaydi. Shu nuqtai nazardan, bilimlarni qayta ishlash va ontologiya kutubxonalarini qurish bir-biridan ajralmasdir. Ontologiyalar kutubxonalarini qurish muammosi boshqa turdagi muammolarda ham paydo bo'ladi, bunda bilimlarni qayta ishlash axborot tizimini yaratishning bevosita maqsadi emas. Masalan, Web-portalni yaratishda foydalanuvchilarga axborot xizmatlarini ko'rsatishning asosiy vazifasi bilan bir qatorda, Internet qidiruv dasturlari uchun ushbu axborot tizimining mazmuni tavsifini qurish masalasini hal qilish kerak. Ushbu tavsif qidiruv robotiga tushunarli bo'lishi uchun etarlicha rasmiy bo'lishi kerak, ya'ni ontologiyadan boshqa narsani anglatmaydi. Bu, albatta, axborot tizimining mazmuni tavsifini qurish vazifasi nafaqat mashina bilimlarini qayta ishlash xizmatini qurishning aniq vazifasi bilan bog'liq holda, balki har qanday tizimni qurishning zarur komponenti sifatida paydo bo'lgan yagona misol emas. funksional maqsadi qayta ishlash vazifasi bilan bevosita bog'liq bo'lmagan axborot tizimi.bilim. Monografiyada ko'rib chiqilgan deyarli barcha axborot tizimlarida ontologiyalarni qurish bitta nazariy asosga, ya'ni predikat hisobiga asoslangan. Asosan, ontologiyalar aniq aksiomalar shaklida ma'lum cheklovlar qo'yilgan predikativ iboralar bilan berilgan bayonotlar to'plamidir. Predikativ iboralar tavsiflangan dunyo haqidagi faktlarni belgilaydi, ya'ni bilimlarni tavsiflash uchun ishlatiladigan predikativ hisob tili birinchi tartibli til edi. Xulosa qilib aytishimiz mumkinki, an'anaviy va eng ko'p qo'llaniladigan

yondashuv ontologiyani birinchi tartibli predikatlar hisobi tili orqali ifodalashga asoslangan bo'lib, unga ikkinchi darajali elementlar ko'pincha qo'shiladi. Masalan, OWL-da siz o'tish qobiliyati yoki funkcionallik kabi xususiyatlarning (predikatlar) ba'zi xususiyatlarini belgilashingiz mumkin. Ikkinchi va undan yuqori darajali tillarning imkoniyatlaridan foydalanish qat'iy cheklangan, bu bilimlarning etarlicha aniq tavsifini va ushbu bilimlarga asoslangan juda tez xulosa chiqarish jarayonlarini ta'minlash zarurati bilan bog'liq.

Predikatlar hisobi yaxshi o'rganilgan sohadir, shuning uchun ontologiya kutubxonasi loyihalarini amalga oshirishning aksariyati ushbu rasmiyatchilikdan foydalanishi tabiiydir. Ammo, predikat hisobidan tashqari, bilimlarni tavsiflashning boshqa yondashuvlari ham mavjud. Shunday qilib, yaqinda bilimlarni tavsiflashning algebraik yondashuvi mashhur bo'ldi. Ushbu yondashuv asosida ontologiya kutubxonalarini qurish tizimlari yaratilmoqda, ular bu yerda bo'sh joy etishmasligi sababli tavsiflanmagan, shuningdek, asosan, ularning yetarli darajada sinovdan o'tkazilmaganligi sababli. Algebraik yondashuvda bilim maxsus shakldagi ifodalar to'plami - atamalar shaklida ifodalanadi. Ekvivalentlik aksiomalari yordamida atamalar shakliga cheklovlar qo'yiladi. Hisoblashlar atamalar to'plami bo'yicha amalga oshirilishi mumkin, bu orqali ontologiyaning turli savollariga javob berish va hisoblash jarayonida yangi bilimlarni yaratish mumkin. Bilimlarni ifodalashga algebraik yondashuvning tavsifi, ehtimol, alohida bobni talab qiladi. Bilim ijtimoiy hodisadir. Bilim odamlar o'rtasidagi kelishuvlar natijasida paydo bo'ladi, ya'ni birgalikda shakllanadi. Kollektiv bilimlarni shakllantirishning bu xususiyati ontologiya kutubxonasini qurish tizimi tomonidan foydalanuvchilarga taqdim etilgan vositalarda aks etishi kerak. Albatta, mavjud tizimlar, ayniqsa ishlab chiquvchilarning etarlicha katta guruhlari jalb qilingan tizimlar bunday vositalarni amalga oshirishi kerak edi. Ammo, o'ziga xos tarzda, ontologiyalarning jamoaviy shakllanish jarayoni o'rganilmagan. Ontologiyalarni jamoaviy shakllantirish vositalarini ta'minlaydigan tizimlarni qurishda qo'llanilishi kerak bo'lgan usullar va yondashuvlarni tavsiflash uchun alohida bob kerak. Ontologiyalar kutubxonalarini qurish tizimlari odatda bilimlarni tavsiflash uchun rasmiy tillarni ifodalaydi, ularni o'rganish va amalga oshirish juda murakkab. Ontologiya kutubxonasida bilimlarni to'liq tavsiflay olish uchun foydalanuvchi

tizimda ushbu bilimlar qaysi tilda oʻrnatilganligini, shuningdek, ushbu tizimda bu bilimlar qanday tamoyillar asosida shakllantirilishini oʻrganishi kerak. Bu ancha mashaqqatli jarayon. Koʻp sonli qiyinchiliklar bilim tavsiflangan tilning ontologik faktlarni qulay tarzda ifodalash uchun sintaktik moslashuvchan emasligi bilan bogʻliq. Kundalik hayotda odamlar yangi faktlarni ifodalash uchun yangi bilimlarni qulay tavsiflash uchun tilning sintaksisi va lugʻatini oʻzgartiradilar. Shu maʼnoda kundalik muloqot tili ochiq. Bilimlarni tavsiflash uchun ochiq tilni amalga oshirish juda mashaqqatli vazifa boʻlib, uni hal qilish maxsus oʻrganish va aniq usullarni qoʻllashni, ham sintaktik tahlilni, ham bunday oʻsib borayotgan bilimlarni modellashtirishga maxsus yondashuvlarni talab qiladi. Shubhasiz, yuzaga keladigan muammolarni toʻliq tavsiflash uchun maxsus bob talab qilinadi. Shunday qilib, bu kitobda hali javob topilmagan koʻplab savollar mavjud. Aytishimiz mumkinki, ushbu kitobda ontologiya kutubxonalarini qurish sohasidagi ishlarning bugungi holatini aks ettiruvchi masalalar yoritilgan. Bu yerda bu boradagi mavjud gʻoyalarni rivojlantirish bilan bogʻliq masalalar muhokama qilinmadi. Muallif tez orada ushbu muammolar tavsifiga bagʻishlangan asar yozilishiga umid qiladi.

FOYDALANILGAN ADABIYOTLAR RO'YXATI

1. Алексеева М. А. Обзор системы Сус. М.:РГГУ (2008). http://ezopproject.wiki.sourceforge.net/Alekseeva_Cус.
2. Клини С. Математическая логика. М.: Мир, 1973.
3. Лондон Дж. Морской волк. М.: АСТ, 2006.
4. Сайт компании Cусcorp. <http://www.cус.com>.
5. Сайт проекта OpenCус. <http://opencус.org>.
6. Страница описания контекстов Cус.<http://www.cус.com/cусdoc/course/contexts-basic-module.html>.
7. Страница проекта Cус в английской Википедии. <http://en.wikipedia.org/wiki/Cус>.
8. Страница описания онтологии Cус. <http://www.cус.com>
9. Страница описания языка CусL на сайте проекта Cус. <http://www.cус.com/cусdoc/ref/cycl-syntax.html>.
10. Страница Cус-NL. <http://www.cус.com/cус/technology/>
11. Чень Ч, Ли Р. Математическая логика и автоматическое доказательство теорем. М.: Наука, 1983.
12. Хювёнен Э., Септянен И. Мир Лиспа. Том 1. Введение в язык Лисп и функциональное программирование. М.: Мир, 1990.
13. Davidson С. Common Sense and the Computer. New Scientist. April 2, 1994.
14. Guha R., Lenat D., Pittman K., Pratt D., Shepherd M. Cус: a midterm report. Communications of the ACM, August 1990/ Vol 33. No 8.
15. Lenat D., Guha R. Building Large Knowledge Based Systems. Addison Wesley, Reading Mass, 1990.
16. Lenat D., Guha R. Enabling Agents to Work Together, Communications of the ACM, July 1994/ Vol 37. No. 7.
17. Lenat D. Steps to Sharing Knowledge. In Toward Very Large Knowledge Bases. Edited by N.J.I. Mars. IOS Press, 1995.
18. Lenat D. Artificial Intelligence. Scientific American, September 1995.
19. Бобров Е. Г. Карл Линней. 1707—1778. — Л.: Наука. 1970. — 285 с.
20. Гоклениус Р. Lexicon philosophicum, quo tanquam clave philisophiae fores aperiunter. Fransofurti, 1613.

21. Драфт стандарта dpANS языка KIF.
<http://logic.stanford.edu/kif/dpans.html>.
22. . Клауберг И. *Metaphysika de ente, quae rectus Ontosophia*. Амстердам, 1991.
23. Когаловский М., Калиниченко Л. Концептуальное моделирование и онтологические модели. В сборнике *Онтологическое моделирования, труды симпозиума в г. Звенигород, 19-20 мая, 2008*.
24. Линней К. Система природы. Царство животных, ч. 1–2. СПб, 1804–1805.
25. Маклейн С. Категории для работающего математика. Перевод под редакцией В. А. Артамонова. М.: Физматлит, 2004.
26. Менделеев Д. И., Периодический закон. Основные статьи, М., 1958.
27. Описание концепций языка RDF на сайте W3.
<http://www.w3.org/TR/rdf-concepts/>
28. Описание языка OWL на сайте W3.
<http://www.w3.org/TR/owlref/>.
29. Описание языка XML на сайте W3.
<http://www.w3.org/XML/>.
30. Сайт Каталога Яндекса. <http://catalog.yandex.ru/>.
31. Сайт компании Cyscorp. <http://www.cys.com>.
32. Сайт Онтолингва.
<http://www.ksl.stanford.edu/software/ontolingua/>.
33. Сайт, посвященный BORO методологии.
<http://www.boroprogram.org>.
34. Сайт проекта OpenCys. <http://opencys.org>.
35. Сайт проекта WordNet. <http://wordnet.princeton.edu>.
36. Сайт языка LISP.
<http://www.lispworks.com/documentation/commonlisp.html>.
37. Сайт KIF. <http://ksl.stanford.edu/knowledge-sharing/kif/>.
38. Страница на сайте консорциума W3, посвященная языку SPARQL. <http://www.w3.org/2001/sw/DataAccess/>.
39. Страница описания онтологии Cys.
http://www.cys.com/cys/technology/whatis_cys_dir/whatdoescysknow
40. Страница описания языка CysL на сайте проекта Cys.
<http://www.cys.com/cysdoc/ref/cycl-syntax.html>.

41. Труды симпозиума «Онтологическое моделирование». М.: ИПИ РАН, г. Звенигород, 19-20 мая, 2008.
42. Буч Г., Рамбо Д., Джекобсон А. Язык UML. Руководство пользователя: Пер. с англ.— М.: ДМК, 2000.
43. Goguen J. What Is a Concept?. ICCS, pp. 52-77, 2005.
44. Goguen J, Harrell F. Foundations for active multimedia narrative: Semiotic spaces and structural blending. *Interaction Studies: Social Behaviour and Communication in Biological and Artificial systems*, 2005.
45. Clancey W. The Knowledge Level Reinterpreted: Modelling SocioTechnical Systems. *International Journal of Intelligent Systems*, 8: 33-49 1993.
46. Codd E. A Relational Model of Data for Large Shared Data Banks. *Commun. ACM* 13(6): pp. 377-387 (1970)
47. Fonteseca F., Martin J. Learning the differences between Ontologies and Conceptual Schemas through Ontology-Driven Information Systems. *Journal of the Association for Information Systems— Special Issue on Ontologies in the Context of IS. Volume 8, Issue 2, Article 3, February 2007.*
48. Gruber T. R. The role of common ontology in achieving sharable, reusable knowledge bases. In J. A. Allen, R. Fikes, and E. Sandewell, editors, *Principles of Knowledge Representation and Reasoning – Proceedings of the Second International Conference*, pp. 601-602. Morgan Kaufmann (1991)
49. Gruber T. R. Ontolingua: A mechanism to support portable ontologies. // Stanford University, Knowledge Systems Laboratory, Technical Report KSL-91-66, Март 1992.
50. Gruber T. R. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *International Journal Human-Computer Studies*, 43(5-6): pp. 907-928. (1995)
51. Guarino N. Formal Ontology, Conceptual Analysis and Knowledge Representation. *International Journal of Human-Computer Studies*, 43(5-6): pp. 625–640. (1995).
52. Guarino N., Formal Ontology in Information Systems. *Proceedings of FOIS'98, Trento, Italy, 6-8 June 1998. Amsterdam, IOS Press*, pp. 3-15.
53. Liskov B. *Abstraction and specification in program development*, MIT Press, Cambridge, MA, 1986.

54. Newell, A. The Knowledge Level. *Artificial Intelligence*, 18: 87-1276 1982.

55. Smith B. Ontology. In Floridi L. (ed.) *The Blackwell Guide to the Philosophy of Computing and Information*. pp. 155-167. 2003.

ESHONQULOV HAKIM ILHOMOVICH

**ONTOLOGIYALAR VA BILIMLARNI IFODALASH,
AVTOMATLASHTIRILGAN AXBOROT TIZIMLARINING
ONTOLOGIYASINI QO'LLASH ASOSLARI**

Muharrir: A. Qalandarov
Texnik muharrir: G. Samiyeva
Musahhih: Sh. Qahhorov
Sahifalovchi: M. Bafoyeva

Nashriyot litsenziyasi AI № 178. 08.12.2010. Original-maketdan bosishga ruxsat etildi: 28.03.2022. Bichimi 60x84. Kegli 16 shponli. «Times New Roman» garn. Ofset bosma usulida bosildi. Ofset bosma qog`ozi. Bosma tobog`i 8,0. Adadi 100. Buyurtma №108.

“Sadriddin Salim Buxoriy” MCHJ
“Durdona” nashriyoti: Buxoro shahri Muhammad Iqbol ko`chasi, 11-uy.
Bahosi kelishilgan narxda.

“Sadriddin Salim Buxoriy” MCHJ bosmaxonasida chop etildi.
Buxoro shahri Muhammad Iqbol ko`chasi, 11-uy. Tel.: 0(365) 221-26-45