IDENTIFICATION OF WEB SERVICES USING EXTENDED METADATA

Идентификация веб-сервисов во время проектирования систем е-услуг

PAPLAŠINĀTU METADATU IZMANTOŠANA WEB SERVISU IDENTIFIKĀCIJĀ

P. Stipravietis, M. Ziema

Atslēgas vārdi: e-pakalpojums, WEB serviss, WEB servisu identificēšana

1. WEB servisu identificēšanas problēmas

Dažādu sistēmu dažādu procesu bieži ir balstīta uz WEB servisu izmantošanu. Šajā rakstā netiek aplūkots gadījums, kad izmantojamie WEB servisi vēl ir jāizstrādā – tiek pieņemts, ka ir pieejams WEB servisu reģistrs, kurā tiek uzglabāti dati par WEB servisiem. Nonākot līdz WEB servisa izmantošanai, bieži rodas problēma – kā no pieejamās WEB servisu kopas izvēlēties piemērotāko uzdevuma veikšanai. Pastāv vairākas pieejas, kā izvēlēties piemērotāko WEB servisu – meklēšana UDDI reģistrā, dažādas paplašinātas UDDI specifikācijas – eUDDI, WEB servisu izvēle atkarībā no to reitinga, kuru ietekmē tā lietotājs ar aģentu palīdzību, dažādas semantiskas pieejas kā SAWSDL un SQUID-WS.

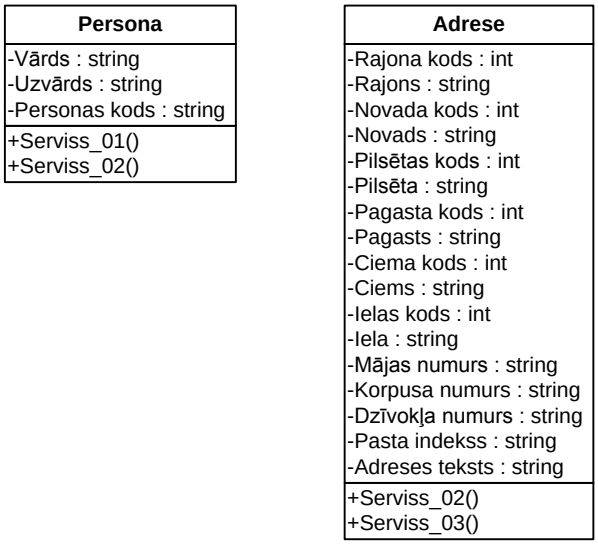
UDDI reģistrs nodrošina meklēšanu pēc WEB servisa izstrādātāja, kategorijas (nozares) vai WEB servisa nosaukuma. Pēc meklēšanas parametru uzdošanas un rezultātu aplūkošanas bieži vien izstrādātājs ir neziņā – atrastā WEB servisu kopa ir liela un katra WEB servisa apraksts var būt nepilnīgs vai nepietiekami izskaidrojošs, vai pat vispār nebūt uzdots. Ar šiem parametriem ļoti reti pietiek, lai varētu atrast sevi interesējošo WEB servisu, tamdēļ kā risinājumu izvēlas papildināt UDDI reģistru ar dažādiem WEB servisu metadatiem, tādējādi paplašinot atlases parametru kopu un samazinot atrasto WEB servisu daudzumu. Šādi metadati var būt, piemēram, WEB servisa izmantošanas intervāls, sasaiste ar citiem WEB servisiem vai dokumentiem u.c.

Lai arī šāda paplašināta metadatu kopa ir noderīga, tā tomēr nedefinē WEB servisa izpildes vispārējo algoritmu un nespēj palīdzēt gadījumā, kad, piemēram, jāatrod WEB serviss, kas kā izejas parametru atgriež personas adresi. Izstrādātājs var nezināt kategoriju, kādā WEB serviss ir iekļauts, tas var nezināt WEB servisa izstrādātāju un tā vārdu – pēc UDDI reģistra nosacījumiem neko par WEB servisu, lai to atrastu. Principā izstrādātājam pat ir vienalga, kas ir publicējis WEB servisu – galvenais, lai tas veic e-pakalpojuma izpildi nodrošinošās darbības. Šādos gadījumos ir vērts ieviest metadatu modeli, kas apraksta WEB servisa ieejas un izejas datus.

2. WEB servisu identificēšana

WEB servisu ieejas un izejas parametri tiek uzdoti kā biznesa vienumi vai to sastāvdaļas, piemēram, Persona – pakalpojuma pieprasītājs, vārds, uzvārds u.c. Šādus parametrus var piekārtot biznesa vienumu hierarhiskajam modelim. Meklēšana datu hierarhijā, par meklēšanas parametriem uzdodot WEB servisa ieejas vai izvades parametru tipus, var atgriezt daudz precīzāku datu kopu. Šajā rakstā meklēšana pēc

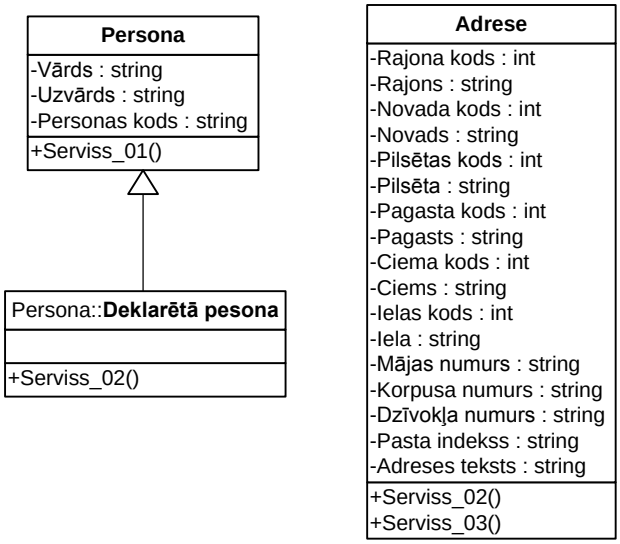
ieejas vai izejas parametriem netiek atsevišķi nodalīta – princips abos gadījumos ir vienāds.



1. attēls – vienkārša datu hierarhija

1. attēlā redzama vienkārša datu hierarhija – persona un adrese. Biznesa vienums „Persona" ir kā parametrs WEB servisiem Serviss_01 un Serviss_02, bet „Adrese" – Serviss_02 un Serviss_03. Līdz ar to kā meklēšanas parametru uzdodot Personu, tiks atgriezti WEB servisi Serviss_01 un Serviss_02; uzdodot Adresi, tiks atgriezti Serviss_02 un Serviss_03, bet uzdodot gan Adresi, gan Personu, tiks atgriezts tikai Serviss_02.

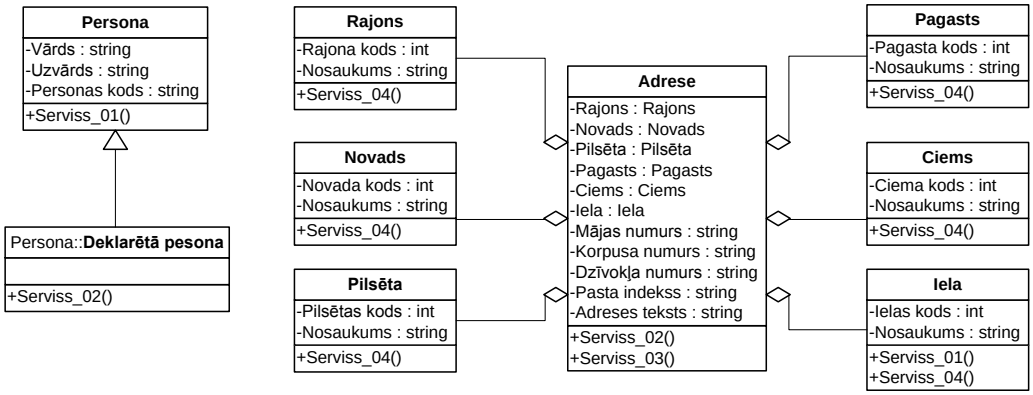
Šāds sadalījums ir ļoti vispārīgs, tamdēļ ir vērts ieviest konkrētākus vienumus, kas manto no vispārīgākiem, teiksim, Deklarētā persona.



2. attēls – datu hierarhija ar mantošanu

2. attēlā redzams biznesa vienums „Deklarētā persona", kas manto no „Personas". Ja arī WEB servisam Serviss_02 parametrs ir uzdots kā „Deklarētā persona", tas tomēr nedrīkst ļaut meklēšanai pēc „Personas" kā rezultātu neatgriezt Serviss_02. Tādējādi var formulēt 1. meklēšanas nosacījumu – meklēšanas rezultāts ir visi WEB servisi,

kas piekārtoti kokam, kura sakne ir uzdotais biznesa vienums, bet bērni – vienumi, kas no tā manto.
Iespējami gadījumi, kad biznesa vienums var sastāvēt no citiem biznesa vienumiem, pēc kuriem arī ir iespējama meklēšana – teiksim, „Adreses" sastāvā ir „Iela". Šādā gadījumā vienumu „Adrese" sauksim par saliktu vienumu, bet „Iela" – par vienkāršu.



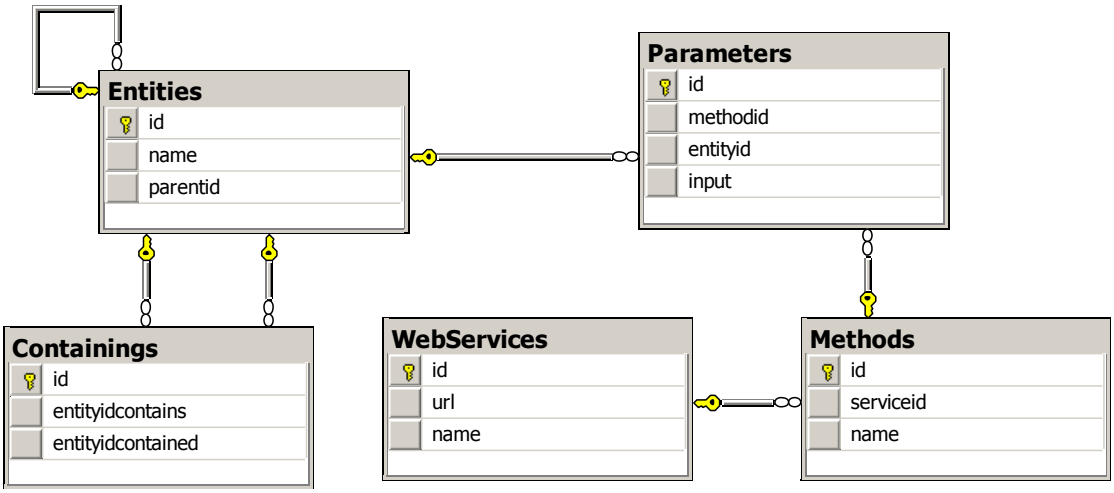
3. attēls – datu hierarhija ar mantošanu un kompozīciju

3. attēlā redzams, no kādiem vienkāršajiem vienumiem sastāv vienums „Adrese". Ja kā meklēšanas parametrs tiek uzdots vienums „Adrese", rezultātā tiks atgriezti WEB servisi Serviss_02 un Serviss_03, bet netiks atgriezts Serviss_04. Ja kā meklēšanas parametrs tiks uzdots vienums „Iela", rezultātā tiks atgriezti visi 4 zināmie WEB servisi – Serviss_01 un Serviss_04 šo vienumu satur atklātā veidā, bet Serviss_02 un Serviss_03 – kā cita vienuma sastāvdaļu. Līdz ar to 2. meklēšanas nosacījums varētu būt šāds – meklēšanas rezultāts ir visi WEB servisi, kuru parametri vai šo parametru sastāvdaļas ir uzdotais biznesa vienums.
Ja meklēšanā tiek uzdoti vairāki parametri, tad rezultāts ir atkarīgs no uzdoto parametru saistības:

$$Q(x \wedge y) = Q(x) \cap Q(y)$$

$$Q(x \vee y) = Q(x) \cup Q(y)$$

Doto meklēšanu var realizēt, WEB servisu reģistru papildinot šādā veidā:



4. attēls – reģistra papildināšana

Tabula „WebServices" satur datus par WEB servisiem, „Methods" – par to metodēm, „Parameters" – par metožu parametriem. Tabula „Entities" satur datus par vienumiem un mantošanu, bet „Containings" apraksta, no kādiem vienumiem sastāv citi vienumi. Tabula „WebServices" vai tās analogs izvēlētajā reģistrā noteikti jau ir, iespējams, ka ir arī tabula, kas satur līdzīgus datus kā „Methods" un daļu no „Parameters".

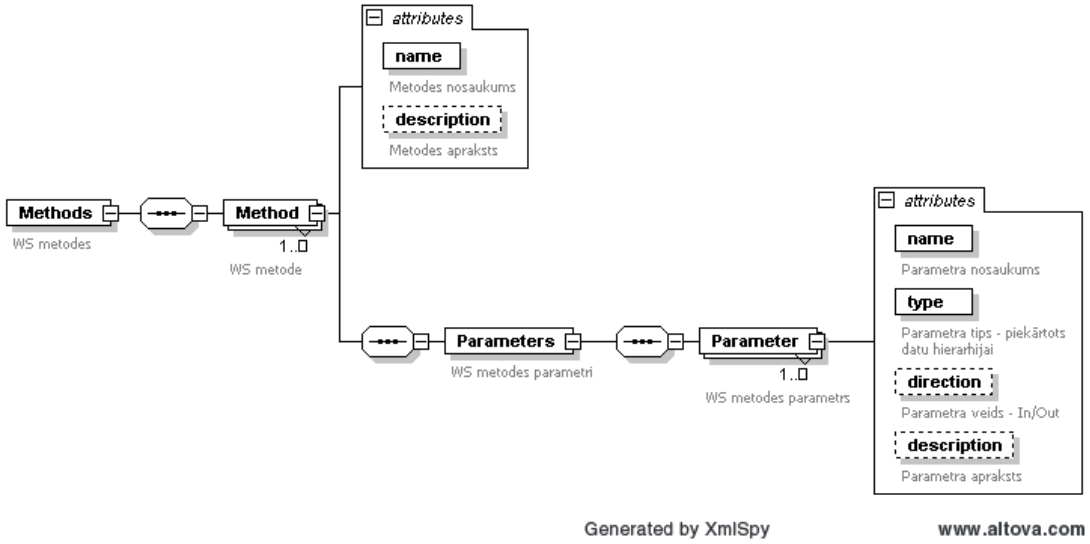
Tabulu „Entities” un „Containings” datus vajadzētu mainīt tikai reģistra turētājam, lai nepieļautu tajās esošo datu kvalitātes kritumu – nevajadzīgus vienumus, dublējošos vienumus u.c. Protams, tas nenodrošinās 100% precizitāti, tomēr samazinās kļūdu rašanās iespējas.
Kad tabulas ir aizpildītas ar datiem, atlasīt WEB servisus, kuru metožu parametri satur uzdoto vienumu, var ar šādu T-SQL pieprasījumu:

```
with EntChildren (id)
as
(
  select E.id from dbo.Entities E where id = @entityID
  union all
  select E.id from dbo.Entities E inner join EntChildren on
    EntChildren.id = E.parentid
),
EntContainers (id)
as
(
  select C.entityidcontains as id from dbo.Containings C where
    entityidcontained = @entityID
  union all
  select C.entityidcontains as id from dbo.Containings C inner join
    EntContainers E on E.id = C.entityidcontained
)
select distinct WS.* from dbo.WebServices WS inner join dbo.Methods M
  on WS.id = M.serviceid inner join dbo.Parameters P on M.id =
  P.methodid inner join EntChildren on P.entityid = EntChildren.id
union
select distinct WS.* from dbo.WebServices WS inner join dbo.Methods M
  on WS.id = M.serviceid inner join dbo.Parameters P on M.id =
  P.methodid inner join EntContainers on P.entityid =
  EntContainers.id
```

Dotajā piemērā `@entityID` – izvēlētā vienuma identifikators.

3. WEB servisu parametru aprakstīšana

Esošo VISS metadatu un e-pakalpojumu identifikācijas standartu iespējams papildināt ar metadatiem, kas apraksta WEB servisā esošās metodes un tajās izmantotos parametrus. Zemāk dots XML shēmas piemērs, kas ļauj veikt aprakstu.



5. attēls – paplašināto metadatu modelis

Šādus metadatus var piekārtot jebkuram reģistrā publicējamam WEB servisam, arī tiem, kuru metožu parametri ir definēti kā „aploksnes" – biznesa operācijai nepieciešamie dati ir uzdoti XML formātā kā atsevišķs padotā parametra atribūts (šāds risinājums ir VISS WEB servisu parametriem ar tipiem `IVISRequest` un `IVISResponse`).
Publicējot WEB servisu un tā ziņojumu shēmas, iespējama automātiska metadatu apstrāde, lai aizpildītu datu tipu hierarhijas tabulas, tomēr jau iepriekš minēto datu kvalitātes apsvērumu dēļ jaunu biznesa vienumu pievienošanu hierarhijai labāk uzticēt WEB servisa reģistra turētājam. Jauno parametru tipu analīzes laikā var atklāties, ka šāds biznesa vienums hierarhijā jau eksistē (tikai ar citādu nosaukumu un definīciju), līdz ar to šādus WEB servisus var vai nu atgriezt atpakaļ izstrādātājam, lai tas pārveido metodes, izmantojot jau publicētas ziņojumu shēmas, vai arī datu hierarhijā ieviest sinonīmu vienumu jēdzienu – izveidojot vēl vienu tabulu, kurā glabāt saites starp vienumiem, kas savstarpēji ir pietiekami līdzīgi (sinonīmu vienumu datu uzkrāšanas gadījumā jāpapildina iepriekš dotais SQL datu atlases pieprasījums). Jebkurā gadījumā lēmums jāpieņem reģistra turētājam.

4. Meklēšana pašreizējā VISS WEB servisu reģistrā

Pašreizējā VISS WEB servisu reģistra versija servisu meklēšanu pēc uzdodamiem parametriem nepiedāvā – tā vietā reģistru ir iespējams pārlūkot Internet pārlūkprogrammā. Pārlūkošana ir bāzēta uz izstrādātāju orientētu hierarhiju – Īpašnieks – tā WEB servisi – servisā izmantotās ziņojumu shēmas. Neskatoties uz daudzajiem metadatiem, šāds risinājums WEB servisu meklētājiem ir vēl nederīgāks kā UDDI reģistrs, kas nodrošina meklēšanu arī pēc kategorijas, ne tikai pēc izstrādātāja.
Pašlaik VISS reģistrā ir publicēti ~20 WEB servisi un ~200 ziņojumu shēmas, kas apraksta tajos esošo metožu parametrus, kā arī nākotnē izmantojamas hierarhijas, līdz ar to daļu no rakstā piedāvātā risinājuma reģistrs jau uztur. Esošā struktūra ļautu meklēt WEB servisu pēc uzdotas XML shēmas (biznesa vienuma), taču tā neatrastu visus hierarhijai netieši piekārtotos WEB servisus, meklējot pēc hierarhijas vispārinājuma – piemēram, meklējot pēc Personas, netiktu atrasta Deklarētā persona. Tāpat netiktu atrasti arī WEB servisi, kuros uzdotais vienums ir apslēpts – uzdots kā cita vienuma sastāvdaļa. Protams, meklēšanas rezultāti ir atkarīgi no sasaistes starp vienumiem un parametriem – manuāli iespējams veikt arī pilnīgi nepareizu sasaisti, tomēr meklēšanas beigās iegūt korektu rezultātu. Ja WEB servisa izstrādātājs ir korekti sagatavojis metadatus, kas atbilst piedāvātajam formātam, WEB servisa publicēšanas brīdī saites iespējams izveidot arī automātiski, līdz ar to meklēšanas rezultāts vienmēr būtu korekts.

5. Kopsavilkums

Sistēmu izstrādātājiem savās sistēmās bieži nākas izmantot WEB servisus. Grūtības rodas, ja uzdevuma veikšanai nepieciešamie servisi nav precīzi zināmi – pieejams tikai reģistrs, kuros tos meklēt. Meklēšana pēc izstrādātāja, kategorijas vai nosaukuma kā standarta UDDI reģistrā bieži vien aizņem daudz laika. Bieži vien gadās, ka izstrādātājs nemaz nezin, kas ir izstrādājis vajadzīgo WEB servisu, kādai kategorijai tas piekārtots, nerunājot nemaz par tā nosaukumu. Šādos gadījumos problēmu var risināt, piedāvājot meklēšanu pēc WEB servisa metožu parametros izmantotajiem tipiem – biznesa vienumiem (personām, adresēm u.c.).

Lai veiktu meklēšanu atbilstoši uzdotajiem parametriem, vispirms jāievieš metadatu struktūra, kas ļautu aprakstīt WEB servisa metodēs esošo parametru tipus. Publicējot WEB servisu ar šādi aizpildītiem metadatiem, tos iespējams apstrādāt un izveidot WEB servisu datus glabājošā datu bāzē saites starp WEB servisiem, to metodēm un izmantotajiem parametriem, kā arī to tipiem. Piedāvātā struktūra ļauj definēt saites starp izmantotajiem tipiem – tie tiek uztverti kā biznesa vienumi, kas var sastāvēt viens no otra (kā Adrese sastāv no Ielas) un mantot viens no otra.
Piedāvāto izmaiņu realizācija sistēmu izstrādātājiem ievērojami atvieglotu WEB servisu identificēšanu.

6. Izmantotā literatūra

1. VISS IS servisu, XML ziņojumu shēmu un e-pakalpojumu katalogs (http://ivis.eps.gov.lv/epr)
2. Integrētā valsts informācijas sistēma, metadatu un e-pakalpojumu identifikācijas standarts
3. UDDI specifikācija (http://www.uddi.org/)
4. SQUID-WS: Semantically Querying the UDDI for discovering Web Services - Preeda Rajasekaran, Rohit Aggarwal, Fred Maier, Matt Ross
5. SAWSDL specifikācija (http://www.w3.org/2002/ws/sawsdl/)
6. Agent-based Architecture for Autonomic Web Service Selection - E. Michael Maximilien, Munindar P. Singh