# LOD Hyödyt ja haasteet – erityisesti tilastollisen datan yhteydessä

JOHDANTO! EXECUTIVE SUMMARY!

## LOD yleisesti

Linked Open Data mahdollistaa tiedon uudelleen käyttämisen mahdollisimman tehokkaasti. Omassa tietoaineistossa pystyy hyödyntämään helposti muita tietoaineistoja. Vastaavasti oman tietoaineiston voi tarjota muiden käyttöön. Koska omaa tietoa avoimesti tarjoava taho ei voi tietää, missä ja miten tätä tietoa tullaan hyödyntämään, on tärkeää, että rajapinnat ja tiedon esitystapa on riittävän yleinen eikä aseta rajoitteita tiedon käyttötavoille.

Yleisenä ja standardoituna tiedon esitystapana RDF on hyvä valinta tiedon esitysmuodolle erityisesti tarkemmin määrittelemättömään käyttöön tarkoitetuissa rajapinnoissa. Verrattuna esimerkiksi XML:ään, se keskittyy itse tietoon ja sen rakenteisiin siinä missä esimerkiksi XML on ensisijaisesti rakenteinen dokumentti ja XML Schema kuvaa dokumentin rakennetta varsinaisen tietomallin sijasta.

RDF on ensisijaisesti abstrakti tietomalli. Sen voi serialisoida useilla eri tavoilla käyttötarkoituksesta riippuen. Yleisiä RDF:n serialisointitapoja ovat RDF/XML ja kompaktimpi sekä luettavampi Turtle. Hyvin määritellyille käyttötapauksille yleinen RDF serialisaatio ja hakutapa on sen sijaan harvoin optimaalinen. Näitä varten kannattaa mahdollisesti tehdä erilliset Web Service rajapinnat – kuten pilotin fasettihaussa.

Olennaista on myös RDF:n tapa identifioida tietoa käyttäen URI-tunnisteita (Uniform Resource Identifier), mikä yleisenä ja yhdenmukaisena tunnisteena mahdollistaa tietoalkioihin viittaamisen missä tahansa yhteydessä. Esimerkiksi tietokantasovelluksissa tieto (rivit) identifioidaan usein tietokannan sisäisellä juoksevalla numerosarjalla, jolloin täysin saman sisältöinenkin tieto saa eri identiteetin kussakin tietokannassa. Kahden osittain samaa tietoa sisältävän relaatiotietokannan yhdistäminen voi osoittautua yllättävänkin haastavaksi siinä missä kaiken RDF aineiston voi suoraan ladata yhteen, ja tiedot yhdistyvät automaattisesti yhteisten URIen ja rakenteiden osalta.

## RDF-tilastojen hyötyjä

PC-Axis –muotoisen aineiston muuttaminen RDF/Scovo-muotoon tekee tiedosta helpommin haettavaa ja ymmärrettävämpää. Scovossa kukin tilastollinen arvo liittyy suoraan sitä luonnehtiviin ulottuvuuksiin, kuten vuosi, paikka ja yksikkö. PX-tiedostomuodossa on vain lista numeroita – tilastollisia arvoja – ja kunkin numeron merkitys aukeaa epäsuorasti laskennallisesti sen position kautta. PX-muoto on hyvä tiedon kompressoinnin ja sitä kautta tallennuksen kannalta, mutta ei ohjelmallisen käyttöön.

Scovo-muodossa saadaan myös erilliset tilastot yhdistettyä helposti niiden käyttämien yhteisten ulottuvuuksien kautta. Monet tilastot liittyvät samoihin paikkoihin, vuosiin, yksiköihin jne. mutta erillisistä tilastoista on työlästä kaivaa esimerkiksi tietyn paikan ja vuoden kaikki tiedot. Scovo-muodosta on yhtä helppoa hakea tietoja erillisten tilastojen yli kuin vain yhden tilaston sisältäkin.

Toisaalta – kuten kaikkea RDF-pohjaista tietoa – Scovo-muotoista tilastotieto voidaan helposti rikastaa. Ulottuvuuksiin, kuten paikkoihin, voidaan helposti lisätä rakennetta (Kallio sijaitsee Helsingissä), koordinaatteja sekä esimerkiksi kuvauksia. Erityisesti RDF mahdollistaa ulkoisten aineistojen linkittämisen tilastotietoihin – ja päin vastoin tilastotietojen linkittämisen ulkoisiin aineistoihin. Aluesarjat –pilotissa linkitimme tilastopaikat Wikipedian RDF-muunnoksen, DBpedian paikkoihin. Näin pystyimme rikastamaan paikkatietoja Wikipedian tiedoilla. Koska RDF:llä tehdyt linkitykset ovat luettavissa kahteen suuntaan, voidaan tilastotietoja vastaavasti hyödyntää muissa DBpediaa käyttävissä sovelluksissa.

Yksi mielenkiintoinen rikastamistapa tilastollisille tiedoille olisi myös dimensioiden muutoksien kuvaaminen kuten Suomen ajallisessa paikkaontologiassa (SAPO). Kun alueet muuttuvat, kaupungit ja alueet yhdistyvät tai jakaantuvat, mitä tehdään historiallisille tilastotiedoille? Tulkitaanko ne uudestaan uuden tilanteen mukaisesti, minkä seurauksena historia vääristyy vai voisiko uudet tiedot yksinkertaisesti kuvata (osittain) uusilla käsitteillä ja tehdä muutos eksplisiittiseksi? Esimerkiksi Helsinki vuonna 2009 tarkoittaa Sipoon liitoksen myötä eri asiaa kuin Helsinki vuonna 2008.

## Haasteita RDF-tilastojen kanssa

Pilotin alkuvaiheen haasteena oli skaalautuvuus. Helppokäyttöinen OpenRDF Sesame Native Store riitti hädin tuskin pilotin alussa valittujen yhdeksän tilaston käsittelemiseen. Tämän jälkeen evaluoimme BigDataa ja Virtuosoa, joista Virtuosolla tehokkuusongelmat ratkesivat helposti. Virtuoson tehokkuus riitti helposti kaikkien Aluesarjat –tilastojen käsittelyyn. Se selviytyi kaikista testatuista kyselyistä hyväksyttävässä ajassa.

Tilastollinen data on luonteeltaan muuttumatonta. Kerran määriteltyä tilastoa ei yleensä tarvitse jälkikäteen muuttaa kuin korkeintaan virheiden korjaamiseksi. Tietojen päivitys ei tarvitse tapahtua reaaliaikaisesti, syöttämällä yksittäisiä arvoja käyttöliittymän kautta, vaan valmis tilasto voidaan ladata palveluun eräajona. Vastaavasti mahdolliset virheet voidaan korjata eräajona. Tilastojen päivitys ei myöskään tarvitse tapahtua transaktionaalisesti, eli että joko koko toiminto onnistuu tai koko toiminto peruutetaan mahdollisen virheen yhteydessä. Kunhan tietokanta itsessään ei korruptoidu (esim. sähkökatkoksen vuoksi), ei haittaa vaikka vain osa tilastosta saataisiin ladattua yhdellä kertaa. Loput voidaan ladata toisella yrittämällä.

Sesamen Native Storagen kanssa ongelmana oli myös se, että ohjelman yllättävä lopettaminen saattoi korruptoida tietokannan. Virtuoso on kuitenkin siinä suhteessa täysiverinen tietokanta, että se pystyy selviytymään tämänkin kaltaisista tilanteista. Vikasietoisuudesta huolimatta kaikista tietokannoista kannattaa ottaa säännölliset varmuuskopiot. Aluesarjojen tapauksessa tämä ei kuitenkaan ole niin olennaista, koska tieto voidaan aina lukea uudestaan alkuperäisistä PX-tiedostoista – kunhan niistä on varmuuskopio.

Tilastollisen aineiston luonteen vuoksi (ei reaaliaikaisia muutoksia, ei tiukkaa transaktionaalisuutta), RDF soveltuu siihen erittäin hyvin. Luonteensa vuoksi se on myös erittäin tehokkaasti hajautettavissa, jolloin mahdollisiin skaalautumisongelmiin pystyy vastaamaan laittamalla lisää “rautaa rajalle”.

## Semantic Web teknologioiden haasteista yleisesti

Ehkä suurin haaste liiketoiminnan kannalta on, että Semantic Web teknologioita osaavia kehittäjiä on vähän. Miten taataan järjestelmän ylläpito pitkässä juoksussa? Toisaalta alamme kehittyy hyvin nopeasti ja uusia teknologioita otetaan käyttöön lähes kaikissa projekteissa; teknologiat vanhentuvat tai muuttuvat radikaalisti hyvin nopeasti. Näin ollen tämä riski ei ole SW:n erityispiirre. Myseman näkökulmasta kyseessä on pitkälti helppokäyttöisten ja alan De-Facto standardeja noudattavien ja edelleen parantavien kirjastojen puute – ja tähän ongelmaan pyrimme itse vahvasti vaikuttamaan esim. RDFBean tuotteellamme, joka on täysin avointa lähdekoodia liiketoimintaystävällisellä lisenssillä.

Yksi potentiaalinen ongelma on SW teknologioiden käyttö väärään tarkoitukseen tai oikeaan tarkoitukseen, mutta väärin. SW-teknologiat ovat hyvä valinta ennen kaikkea data-integraatioon, mistä Aluesarjoissa on kysymys, mutta eivät ratkaise kaikkia maailman ongelmia. Käyttäminen väärin perustein, puutteellisella osaamisella ja/tai väärään tarkoitukseen, voi aiheuttaa tilanteen, jossa “yksinkertainen asia tehdään monimutkaisesti”. Tämäkin ongelma on tosin yleismaailmallinen, eikä millään tavoin SW-teknologioiden erityispiirre.

Yksi riski on myös se, että SW on yhä vahvasti kehittyvä teknologiaperhe. Tietyt kypsemmissä teknologioissa (esim. relaatiokannat) itsestään selvät toiminnot eivät ole RDF-puolella vielä valmiita. Esimerkiksi tietojen manipuolointikieli SPARQL Update –kieli (vrt. SQL insert into ja update) ei ole vielä valmis saati yleistynyt. Myös keskeneräisessä SPARQL 1.1 versiossa tulevat aggregaattioperaatiot (esim. summa, keskiarvo) eivät ole vielä yleistyneet. Virtuoso on siitä hyvä valinta, että se tukee sekä Update-kieltä että aggregaatteja.

Yksittäisetkin tilastot ovat huomattavan suuria RDF-muodossa. Kokonaisten tilastojen serialisointi esim. RDF/XML-muotoon ei ole järkevää, vaan tilastot on vietävä suoraan skaalautuvaan RDF-tietokantaan. Tietokannan on kyettävä käsittelemään tehokkaasti satojakin miljoonia triplejä.