

Relaatiotemporaalinen tietomalli mittaustiedon hallintaa varten

*Antoni Wolski
Johannes Arminen
Antti Pesonen*

VTT Tietotekniikka, PL 1201, 02044 VTT, Finland
Puh. (09) 4561, Fax. (09) 456 6027
sähköposti: {antoni.wolski, johannes.arminen, antti.pesonen}@vtt.fi

AVAINSANAT aikasarja, temporaalinen tietomalli, prosessinhallinta, mittaus

1. JOHDANTO

Teollisuusprosesseissa syntyy suuria määriä tietoja, joita hyödynnetään automaatiojärjestelmissä ja prosessinhallintajärjestelmissä. Tiedon määrän kasvaessa ja prosessinhallintaohjelmien monimutkaistessa tiedon joustava käyttö on yhä tärkeämpää. On tiedonhallintakeinoin taattava, että prosessissa syntynyt tai jalostettu tieto on tallessa niin, että erilaisia, toisistaan poikkeavia sovellusten tietotarpeita voidaan tyydyttää tehokkaasti ja ohjelmoinnin kannalta helposti. Perinteisillä tietotekniikan sovellusalueilla tietokannanhallintajärjestelmiä on käytetty tähän tarkoitukseen. Joustavan, sovellustarpeisiin muokkautuvan käytötavan lisäksi ne takaavat tiedon eheyden ja pysyvyyden erilaisten rinnakkais- ja toipumistekniikoiden avulla. Perinteiset tietokantaratkaisut (jotka nykyisin pääosin perustuvat SQL-kieleen [ISO92]) eivät kuitenkaan sovi hyvin teollisten prosessinhallintajärjestelmien pohjaksi. Yksi syy on SQL-tietokantojen pohjana oleva relaatiomalli, joka ei sovellu luontevasti ajallisten ilmiöiden mallintamiseen. Tarvitaan uusi tietomalli, joka sopisi paremmin prosessitiedoissa esiintyvien aikasarjojen esittämiseen. Toisaalta SQL-kielen vakiintunut asema tietokanta-alueella vaatii, että uusi malli on yhteensopiva SQL-kielen kanssa ja erityistoiminnot on turvattava SQL-kielen laajennusten avulla. VTT Tietotekniikassa kehitetty RQL-kieli ja sitä vastaava tietomalli on kehitetty yllä mainitulla tavalla. Uusi tietomalli on toteutettu RapidBase-nimisessä aikasarjatietokantaohjelmistossa, joka sopii pohjaksi moneen tosiaikaisiin prosessitietoja hyödyntävään sovellukseen.

Luvussa 2 analysoidaan tarkemmin aikasarjan rakenne ja ja tietokannanhallintajärjestelmältä vaaditut ominaisuudet. Luvussa 3 tarkastellaan erilaisia tapoja esittää aikasarjat relaatiotietokannoissa ja näihin tapoihin liittyvät ongelmat. Luku 4 sisältää muiden kehityssuuntien yhteenvedon. Luvussa 5 esitetään RQL-kieli ja -tietomalli. RQL-esimerkkien avulla demonstroidaan tietomallin ominaisuuksia.

2. AIKASARJAN ANATOMIA

Koska prosessin yksittäiset mittaukset (ja usein myös niistä johdetut tiedot) ovat sidottuja aikaan, tiedoista muodostuu aikasarjoja, joilla on seuraavat yhteiset ominaisuudet:

Tietopiste on rakenteellinen tieto-olio, joka esittää reaali maailman mittauspistettä. Osa tietopistettä kuvaavista tiedoista on ajasta riippumattomia, kuten nimet, mittauslaitteen tyyppimerkinnot ja tekniset tiedot.

Aikasarja esittää jotakin ajassa mitattua tietopisteen suuretta tai suureita. Aikasarjassa olevia tietoja yhdistää kronologinen järjestys.

Aikasarjan tietue kertoo yksittäisestä mittauksesta. Jokaisen tietueen yhteydessä on olemassa tieto mittaustajasta ja mahdollisesti tietueen voimassaoloajasta (valid time). Tietueeseen kuuluu yksi tai kaksi aikaleimaa, riippuen toteutustavasta.

Aikasarjan tietueen alkio on tietueen elementti. Se esittää mitatun arvon.

Tietopistetyyppi on valmis kaava, jonka mukaan voidaan luoda uusia tietopisteitä aikasarjoihin.

Tietopisteryhmä on kokoelma tietopisteitä, jotka yhdistää jokin tekijä.

Aikasarjatietokanta on joukko tietopisteitä tyyppineen, ryhmineen ja aikasarjoineen. Sitä hallitaan ja käsitellään sopivan ohjelmiston avulla.

Tyypillisin aikasarjatietokannan toiminto on voimassaoloajan perusteella tapahtuva liitos (eli voimassaoloaikaliitos, valid time join). Sellainen tarvitaan, kun esimerkiksi halutaan tietää, *mikä oli prosessin tila ajanhetkellä t_1* . Silloin ei riitä, että järjestelmä tietää mittausravon mittaushetken – tiedossa pitäisi olla myös hetki johon asti mittausta on voimassa. Silloin vasta voidaan muodostaa tulos kaikista mittausravoista, joiden voimassaoloaika sisältää kyseisen ajanhetken t_1 .

Aikasarjojen hallintajärjestelmälle voidaan asettaa muitakin vaatimuksia, kuten että mittausravot interpoloidaan mittaushetkien välillä, koostearvot (esim. keskiarvot) lasketaan automaattisesti ja vanhentuneet tietueet poistetaan. Aikasarjat voivat olla myös ainakin kahta eri tyyppiä. Säännöllisessä sarjassa mittausravot ovat säännöllisin aikaväleihin. Epäsäännöllisessä sarjassa mittausravot voivat olla mielivaltaisia.

Aikasarjakyselykieli tarvitaan ja aikapohjaisten toimintojen (kuten voimassaoloaikaliitoksen) optimointi on tärkeää.

Yleiskäyttöisen aikasarjahallinnan tarpeet olivat hyvin pitkään vailla niille kuuluvaa huomiota. Tutkimusosalalla on herätty ongelmaan vasta viime aikoina [Dre94]. Viime vuosina on kaupallisiin tietokantajärjestelmiin tuotu joukko aikasarjalajennuksia, jotka eivät kuitenkaan tue hyvin mittausravon tallennustarpeita. Ensin pohditaan, miksei relaatiotietokanta taivu tähän tarpeeseen.

3. TAIPUMATON RELAATIOTIETOKANTA

Aikasarjan tuominen relaatiokantaan on kivuliasta vaikei mahdotonta.

Esimerkiksi seuraava taulu esittää lämpötila-antureiden mittauksia. Taulun joka rivillä on mittaushetken aikaleima.

Probe_id	Time	Tempr
TEMP12	10:20	24
TEMP34	12:10	99
TEMP12	13:10	20

On selvää, että aikaleima kertoo voimassaoloajan alkuhetken, kuten rivillä (TEMP12, 10:20). Mutta missä kerrotaan voimassaoloajan loppuu? Se löytyy jollakin toisella rivillä, joka esittää samaan yksilöön kohdistuvan seuraavan muutoksen, eli tässä tapauksessa rivillä (TEMP12, 13:10).

Kun halutaan suorittaa eri mittaustaulujen voimassaololiitos, tulosrivi muodostetaan voimassaoloajaltaan leikkaavista lähtötaulukien riveistä. Miten sitten kirjoitetaan yllä olevan taulutyypin mukaisista tauluista lähtevä SQL-kysely? Ei mitenkään, koska SQL ei tunne käsitettä *ajassa seuraava rivi*.

Ratkaisuna on koko voimassaoloaikatiedon kirjoittaminen joka riville, kuten seuraavassa esimerkissä.

Probe_id	Beg	End	Tempr
TEMP12	10:20	13:10	24
TEMP34	12:10	NULL	99
TEMP12	13:10	NULL	20

NULL-arvot tulkitaan siten, että tieto on voimassa nykyhetkelläkin.

Täten rakennettujen taulujen voimassaololiitos onnistuu nyt yhdellä SQL-lauseella, mutta siitä tulee monimutkainen. Voidaan myös perustellusti epäillä, kuinka hyvin tietokantatuotteen optimoija selviää tehtävästään.

Vaikka ratkaisu toimii, siinä on muitakin ongelmia. Aikatiedot on esitetty tuplasti ja tietojen päivityksissä täytyy noudata tiukkoja sääntöjä: rivin lisäämiseen tai muuttamiseen liittyy aina jonkun toisen rivin muuttamista. Taulun aikaeheys voi helposti rikkoontua, eikä sitä voi ylläpitää automaattisesti järjestelmän toimesta. Lähteestä [Gol93] löytyy muitakin keinoja, mutta relaatiomallin kankeudesta ei niiden avulla pääse eroon.

4. MUUT LÄHESTYMISTAVAT

Tietokantakirjallisuutta tuntevat lukijat tietävät, että temporaaliset tietokannat ovat tulossa. Tietokantatutkijoiden yhteistyönä luotu kieli TSQL2 on kuvattu kirjassa [Sno95] ja useissa artikkeleissa, kuten [Sno94]. TSQL2 on otettu pohjaksi standardi SQL-kielen laajennuksessa, jonka nimi on *SQL-Temporal*. Nykytiedon mukaan laajennuksen standardi valmistuisi vuonna 2001. Kaupallisia toteutuksia ei vielä ole ja aika näyttää tuleeko niitä.

TSQL2-pohjainen tietokantajärjestelmä ratkaisisi ylhäällä esitetyt ongelmat, mutta puutteita on silti: toisaalta aikasarjakäsittelyn kannalta TSQL2-kielessä on tarpeetonta painolastia ja toisaalta joitakin ominaisuuksia puuttuu, esim. aikasarjatietueen käsite ja eri tyyppisten aikasarjojen tuki.

Tietokantaohjelmistojen valmistajat IBM, Informix ja Oracle ovat tuoneet tuotteisiinsa aikasarjat laajennustietotyyppin muodossa. Toteutukset on suunnattu finanssimarkkinoiden sovelluksiin. Mittausaikasarjoja voidaan niiden avulla toteuttaa, mutta käyttö on hankalaa ja tehotonta. Yhtenä haittana on tarve turvautua erityisiin funktioihin, jotka ovat erillään itse tietokantakielestä.

Teollisten valvomo-ohjelmistojen valmistajat ovat aina tukeneet mittausaikasarjoja jossakin muodossa. Tällaisia tuotteita ovat mm. Wonderwaren *InTouch*, National Instrumentsin *LabVIEW* ja Gensymin *G2*. Näillä välineillä on kuitenkin vaikea rakentaa yleiskäyttöinen aikasarjatietokanta. Pisimmällä on Wonderware *IndustrialSQL Server* -tuotteella, jossa Microsoftin SQL Server on laajennettu aikasarjarakenteilla. Rakenteet ovat kuitenkin hyvin yksinkertaisia (tietopistetyypit ja aikasarjatietueet puuttuvat), eikä aikasarja-alkiota voi muuttaa luomisen jälkeen.

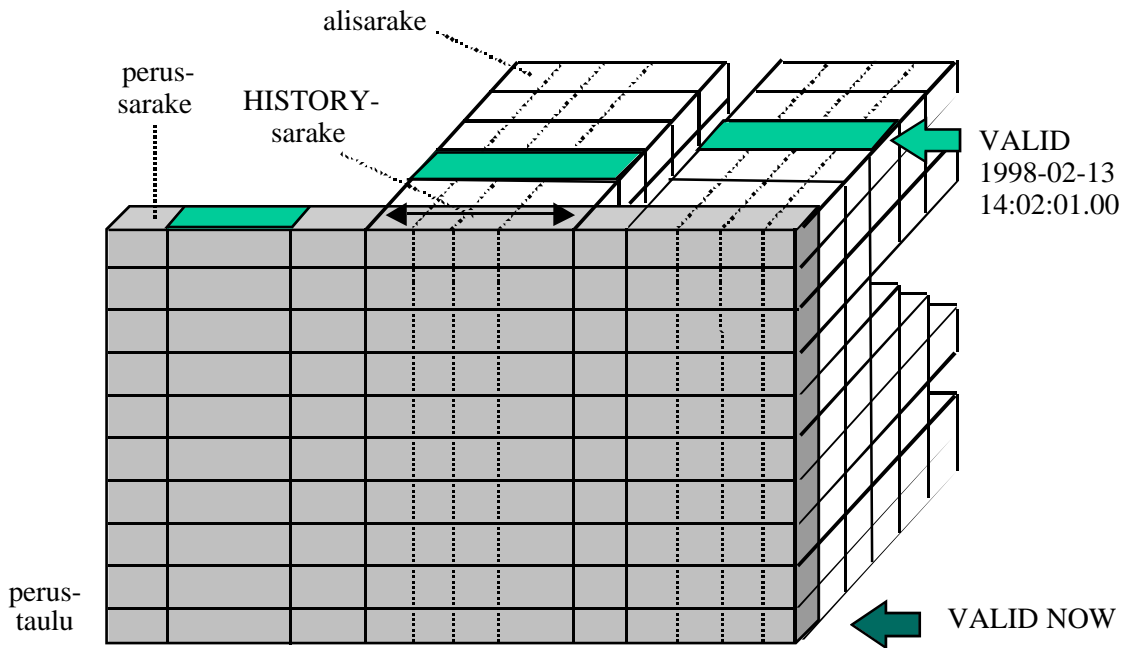
5. RQL-KIELI ja -TIETOMALLI

Olemme VTT Tietotekniikassa yrittäneet luoda helpotajuksen ja tehokkaasti toteutettavan aikasarjatietomallin. RapidBase-aikasarjatietokantajärjestelmässä on toteutettu RQL-kieli (RapidBase Query language) ja sen pohjalla oleva laajennettu relaatiomalli.

Kuva 1 kertoo, miten esitetään prosessimallin olioita RQL-taulujen avulla. Yksi perustaulu vastaa tietopisteen tyyppiä ja rivi yhtä tietopistettä. Esimerkiksi kaikki lämpötila-anturit voitaisiin esittää yhdessä perustaulussa. Jos halutaan lisätä yksi anturi prosessimalliin, taulun lisätään yksi rivi.

Tietopisteen pysyvät tiedot on esitetty tavallisilla perussarakkeilla, kuten perinteisessä tietokannassa. Sen sijaan jokainen tietopisteeseen liittyvä aikasarja esitetään erityisellä HISTORY-tyyppisellä sarakkeella. Tällaisen sarakkeen arvona on kronologisesti järjestetty aikasarjatietueiden sekvenssi. Aikasarjatietue jakautuu alisarakearvoihin, jotka ovat rivin esittämisen olion tietyn hetken tilaa kuvaavia tietoja. Jokaisella tietueella on sisäänrakennettu aikaleimatieto, jonka perusteella järjestelmä päättää tietueen voimassaolon ajasta..

HISTORY-tyypin arvoilla eli aikasarjoilla on ominaisuudet, jotka puuttuvat relaatiotietomallista. Järjestelmä ylläpitää tietueiden järjestystä, aikaleimaa tietueet automaattisesti (ellei sovellus anna aikaleimaa) ja takaa historian maksimaalisen pituuden poistamalla vanhennetut tiedot tietokannasta. Jos vanhentuneet tiedot halutaan kuitenkin säilyttää, tietokantaan määritellyt arkistoijaoliot pitävät huolen tietojen arkistoinnista toiseen taltioon.



Kuva 1. RQL-tietomallin mukainen perustaulu (esimerkki).

RQL-malli supistuu helposti tavalliseen relaatiomalliin, jos vain nykyinen tila on käsittelyn kohteena. RQL-perustaulu voidaan silloin nähdä SQL-perustauluna (kuvan 1 etualalla oleva tummennettu osa). Tavalliset SQL-komennot kohdistuvat tähän 'nykynäkymään', johon kuuluvat aikasarjojen kronologisesti viimeiset tietueet. Laajennetun kieliopin avulla pystytään myös helposti päivittämään ja hakemaan aikasarjoissa olevia tietoja.

Kahden historian perustaulu voidaan luoda esimerkiksi seuraavalla RQL-lauseella:

```
CREATE TABLE tempr_probes (
  probe_id CHAR(8),
  name     VARCHAR(80),
  type     CHAR(10),
  scale    INT,
  measur_h HISTORY (
    tempr INT,
    quality TINYINT
  ) SIZE 10000
  state_h  HISTORY (
    state CHAR(8)
  ) SIZE 1000
```

Tässä esimerkissä taulun historiasarakkeista toinen on tarkoitettu mittausarvojen tallentamiseen ja toinen anturin tilan (ON/OFF) seuraamiseen. On prosessimalliin suunnittelijan tehtävä valita mistä olion ominaksista halutaan historiatietoja talteen ja mistä ei.

Uuden anturin lisääminen prosessimalliin käy seuraavalla lauseella:

```
INSERT INTO tempr_probes
  (probe_id, name, type, scale, state_h.state)
VALUES
  ('TEMP34', 'Axis head tempr meter', 'TM2', 10, 'OFF')
```

Yllä oleva lause alustaa uuden rivin perussarakearvot ja yhden historian tilaa. Historiaan *state_h* on kirjoitettu arvo 'OFF' ja historiatietuen automaattinen aikaleimä kertoo arvon luomisen hetkestä. Toinen historia on vielä kuitenkin alustamaton ja se arvo (ja myös alisarakeiden arvo) on NULL, eli 'tyhjä'.

SQL-tietokannoissa sarakearvojen muuttaminen tapahtuu UPDATE-lauseella. Jos RQL-kielessä UPDATE-lause kohdistetaan historian alisarakeeseen, sen nykyäkymän mukainen arvo muuttuu ja vanha arvo siirtyy historiaan. Uuden anturin tila muutetaan näin:

```
UPDATE tempr_probes
  SET state_h.state = 'ON'
  WHERE probe_id = 'TEMP34'
```

Mittausarvoja syötetään vastaavalla tavalla. Tässäkin tapauksessa käytetään UPDATE-lauseita, koska kyseessä on nykyäkymän päivitys:

```
UPDATE tempr_probes SET
  measur_h.tempr = 124,
  measur_h.quality = 3
  WHERE probe_id = 'TEMP34'
```

Tähän asti esitetyt INSERT ja UPDATE-lauseet ovat SQL-standardin mukaisia. Jos SELECT-lause esitetään standardinmukaisella tavalla, sekin kohdistuu RQL-taulun nykyäkymään. Esimerkiksi lause

```
SELECT probe_id, name, measur_h.tempr
  FROM tempr_probes
  WHERE state_h.state = 'ON'
```

tuottaa tulostaulun, jossa on ON-tilassa olevien antureiden tunnistiedot ja tuoreimmat mittaukset.

RQL-kieli tuo laajennukset, joiden avulla historiatiedot voidaan ylläpitää ja hakea. Esimerkiksi vanhoja historiatietoja päivitetään UPDATE HISTORY -lauseella.

Historiatiedot haetaan tavallisella SELECT-lauseella, jossa on jokin historiaan viittaava temporaalinen predikaatti. Temporaalisista predikaateista kätevin on VALID-predikaatti, jolla annetaan haettavien tietojen voimassaoloaika. RQL-kielen SELECT-lauseessa VALID-predikaatti on aina mukana vaikka käyttäjä sitä ei antaisi. Sen oletusarvomuooto on VALID NOW, joka viittaa nykyäkymään (kuva 1). VALID-predikaatissa voidaan antaa aikapiste, jolloin haetut tiedot ovat voimassa. Esimerkiksi lause

```
SELECT probe_id, name, state_h.state, measur_h.tempr
  FROM tempr_probes
  WHERE VALID '1998-02-13 14:02:01.000'
```

tuottaa listan kaikista antureista ja niiden tiloista ja mitatuista arvoista mainitulla hetkellä. VALID-predikaatilla on useita aikapisteen ja -jaksojen määrittelyyn sopivia muotoja. Esimerkiksi näin haetaan viimeisen minuutin mittaustietoja:

```
SELECT ots, probe_id, name, measur_h.tempr
  FROM tempr_probes
  WHERE VALID FROM NOW - INTERVAL '1' MINUTE
```

Sarake *ots* (object timestamp) on tulostaulun virtuaalinen sarake ja se kertoo mittauksen ajankohdan. Se sisällytetään aina tulostauluun, jos tuloksessa on historiatietoja.. Sen arvo on laskettu historiatietuiden aikaleimatiedoista.

Kun kaksi tai useampi historia yhdistetään yhteen tulostauluun, järjestelmä suorittaa historioiden voimassaololiitoksen niin, että kussakin tulostaulun rivissä on samassa aikajaksoissa voimassa olevia tietoja.. Seuraavaan lauseen kahden historian voimassaololiitoksen tuloksessa jokaisen tulosrivin voimassaoloajan kertovat sarakkeet *ots* ja *ots_end* (viimeksi mainittu on myös virtuaalisarake)

```
SELECT ots, ots_end, probe_id, state_h.state, measur_h.tempr
  FROM tempr_probes
  WHERE VALID BEFORE NOW
```

Usein tarvitaan tietoa säännöllisin aikavälein voimassa olevista arvoista (esim. trendinäytön tuottamiseen). RQL-kielen aikapisteliitos (timepoint join) tuottaa tarvittavan tuloksen. Se voidaan kohdistaa samassa lauseessa muodostettuun voimassaololiitokseen, kuten esimerkiksi tässä lauseessa, joka tuottaa 10 sekunnin aikavälein tila- ja mittaustietoja yhden tunnin ajalta.

```
SELECT ots, probe_id, state_h.state, measur_h.tempr
FROM tempr_probes
TIMEPOINT SERIES INTERVAL '10' SECOND
WHERE VALID FROM NOW - INTERVAL '1' HOUR
```

Tässä kuvattu RQL-kieli ja -tietomalli on toteutettu VTT Tietotekniikan RapidBase-järjestelmässä¹. Sovellusohjelmat käyttävät RapidBase-tietokantaa lähettämällä RQL-lauseita RapidBase-palvelimeen tavanomaisten SQL-tietokannoista tuttujen etäkäyttöliittymien kautta, kuten ODBC ja JDBC. Oma RapidBase C++-pohjainen liittymäluokkakirjasto on myös käytettävissä.

Tässä esitetyn tietomallin lisäksi RapidBase-järjestelmän erikoisuuksiin kuuluvat muunmuassa keskusmuistipohjainen tietokannan toteutus ja ilmaisuvoimainen laukaisinjärjestelmä, johon kuuluvat myös sumean päättelyn pohjalta toimivat laukaisimet.

6. LÄHDELUETTELO

- [Dre94] W. Dreyer et al.: Research Perspectives for Time Series Management Systems. SIGMOD Record, 23(1), March 1994, sivut. 10-15.
- [Gol93] R. Goldring: An Introduction to Time-Series Data. InfoDB, 7(2), Spring 1993, sivut 22 - 25
- [ISO92] ISO/IEC 9075. Information processing systems - Database language SQL. International standard, third edition, 1992. Ref. No. ISO 9075 : 1992 (E).
- [Sno94] R. Snodgrass et al.: TSQL2 Language Specification. ACM SIGMOD Record, 23(1), March 1994, sivut 65-86.
- [Sno95] R. Snodgrass (ed.): The TSQL2 Temporal Query Language. Kluwer Academic Publishers, 1995, 674 s.

¹ <http://www.vtt.fi/tte/projects/rapid/>