Dynaamiset kyselyt

Jan Lindström

Helsinki 16. maaliskuuta 1998

HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos
Tieteellisen kirjoittamisen kurssi syksy 1995
Tutkielma

Sisältö

1 Johdanto

Useimmat tietokantajärjestelmät vaativat käyttäjiltään monimutkaisten kyselyiden luomista, mikä edellyttää käyttäjältä tietokannan loogisen rakenteen tuntemusta. Tietokantakyselyt muodostetaan yleensä korkean tason kyselykielellä kuten SQL:llä tai QUEL:lla. Kyselykielen oppiminen voi kestää aloittelijalla kauan. Siksi tiedonhaussa on selvästi tarvetta helppokäyttöisiin, nopeisiin ja tehokkaisiin kyselymenetelmiin [AWS92].

Shneidermanin [Shn83] mukaan *suorakäsittelyssä* (direct manipulation) toiminnan kohteet ja toiminnot ovat jatkuvasti näkyvissä. Toiminnot suoritetaan osoittamalla, siirtämällä tai painamalla painonappia. Toiminnot toteutetaan välittömästi ja ne ovat yleensä peruttavissa. Suorakäsittely on osoittautunut onnistuneeksi toteutustavaksi useissa ohjelmistoissa. Tällaisia ohjelmistoja ovat esimerkiksi editorit, taulukkolaskentaohjelmat, tietokoneavusteiset suunnitteluohjelmat, tietokonepelit ja graafiset käyttöjärjestelmäympäristöt, kuten Apple Macintosh.

Suorakäsittelyn soveltamista tietokantoihin on yritetty vain muutamia kertoja [AWS92]. Zloof [Zlo75] esittelee suorakäsittelyn käyttöä tiedonkäsittelyyn Query by Example -järjestelmässä, joka perustuu relaatioiden suoraan esittämiseen näyttöruudulla. Toinen yritys luoda käyttäjäystävällisempi tietokanta suorakäsittelyä käyttäen on PICASSO-kyselykieli [KKS88]. PICASSO:ssa kyselyt voidaan syöttää hiiren ja valikoiden avulla, mutta käyttäjä joutuu suorittamaan useita operaatioita jokaisessa kohdassa. Kumpikaan näistä esimerkkisovelluksista ei yhdistä graafista syöttöä ja tulostusta [AWS92].

Dynaamiset kyselyt (dynamic queries) kuvaavat käyttäjän interaktiivista hallintaa visuaalisiin kyselyparametreihin [Shn94]. Dynaamisilla kyselyillä käyttäjä luo kyselyitä, jotka tuottavat nopeita tietokantakyselyn tuloksia visuaalisella esityksellä elävöitettynä. Dynaamiset kyselyt soveltavat suorakäsittelyn periaatteita tietokantaympäristössä. Dynaamisten kyselyjen ideana on, että kun käyttäjä selailee informaatiosivuja, hän samalla täydentää kyselyä ja näkee jatkuvasti muuttuneet kyselyn tulokset.

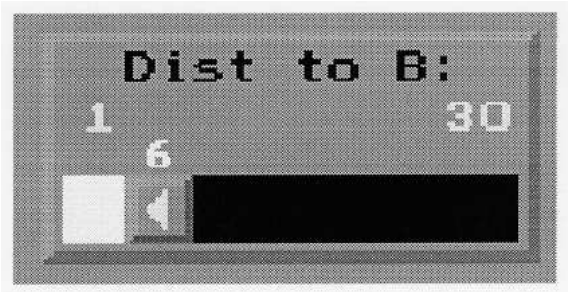
Dynaamisten kyselyiden käyttöliittymä toteutetaan suorakäsittelyn periaatteita käyttäen esittämällä kyselyt, tietokanta ja kyselyn tulokset graafisessa muodossa [AWS92]. Dynaamisissa kyselyissä kyselylle asetetaan selvästi näkyvät rajat. Kyselyn tulokset esitetään välittömästi jokaisen kyselyn muutoksen jälkeen.

Tässä tutkielmassa esitetään, kuinka dynaamisten kyselyiden käyttöliittymä voidaan yleisesti toteuttaa ja arvioidaan dynaamisten kyselyiden etuja ja ongelmia. Tässä tutkielmassa esitetään myös kuinka dynaamisia kyselyitä pystytään laajentamaan käyttäen suodatintoimintoja. Suodatintoiminnoista esitetään peruskäsitteet ryhmittely ja toteutus. Suodatintoimintoja laajennetaan soveltumaan myös reaaliarvoisiin kyselyihin ja puuttuvan tiedon esitykseen.

2 Dynaamiset kyselyt

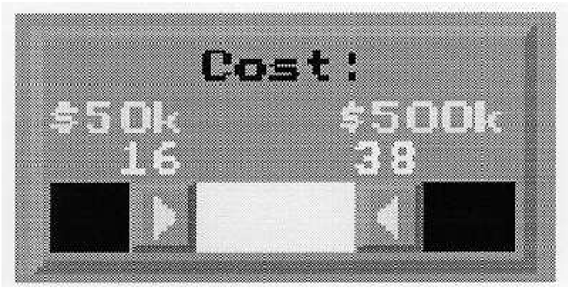
2.1 Dynaamisten kyselyiden toteutus

Dynaamisissa kyselyissä kysely esitetään erilaisilla graafisilla *käyttöliittymäolioilla* (widgets) [AWS92]. Tällainen käyttöliittymäolio on esimerkiksi *liukusäädin* (slider). Liukusäädin (kuva 1) sisältää *siirtolaatikon* (drag box), kaksi arvoa, jotka kertovat liukusäätimen minimi- ja maksimiarvot, sekä valitun arvon. Esimerkiksi kuvan 1 liukusäädin kuvaa arvoja, jotka ovat pienempiä tai yhtä suuria kuin arvo 6.



Kuva 1: Liukusäädin [WiS92].

Liukusäätimessä voi olla myös kaksi siirtolaatikkoa, jolloin sillä kuvataan kahden valitun arvon väliä (kuva 2). Kyselyn tuloksena ovat tällöin ne oliot, joissa attribuutin arvo on suurempi tai yhtä suuri kuin vasen arvo ja pienempi tai yhtä suuri kuin oikeanpuoleinen arvo.



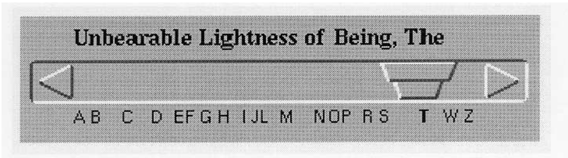
Kuva 2: Liukusäädin, joka kuvaa arvoväliä [WiS92].

Toinen tapa esittää kyselyn parametri on käyttää *painonappia* (button). Painonappia voidaan käyttää parametreihin, joissa kyselyn attribuutti on tyypiltään totuusarvotyyppinen. Esimerkiksi Dynamic HomeFinder -järjestelmässä [WiS92] asunnoista on tietokannassa tieto, onko asunnossa parveketta. Parveketieto voidaan toteuttaa käyttöliittymässä painonapilla, koska tiedolla on vain kaksi mahdollista arvoa. Myös muita kuin totuusarvoisia kyselyparametrejä voidaan toteuttaa painonapilla. Tällaisia toimintoja ovat esimerkiksi <, >, MIN, MAX, AND ja OR.

Yhdeksi dynaamisten kyselyiden toteutusongelmaksi ovat osoittautuneet merkkijonon sisältävät attribuutit. Esimerkki tälläisesta attribuutista on elokuvan nimi, jota haluttaisiin käyttää hakukriteerinä. Eräs mahdollisuus toteuttaa merkkijonoattribuutit on käyttää *aakkosel-*

lista liukusäädintä (alphaslider) [OLS93, AhS94a, AhS94b]. Aakkosellinen liukusäädin voidaan toteuttaa muuten kuten normaali liukusäädin, mutta liukusäätimen yläpuolella lukee valittu arvo ja alapuolella aakkosten indeksi (kuva 3).

Kuvan 3 toteutuksessa aakkoselliseen liukusäätimeen on lisätty molempiin päihin painonapit, joilla voidaan selata arvoja yksitellen haluttuun suuntaan. Hieman kuvasta 3 poikkeavaa toteutusta on käytetty esimerkiksi FilmFinder-järjestelmässä [AhS94b].



Kuva 3: Aakkosellinen liukusäädin [AhS94a].

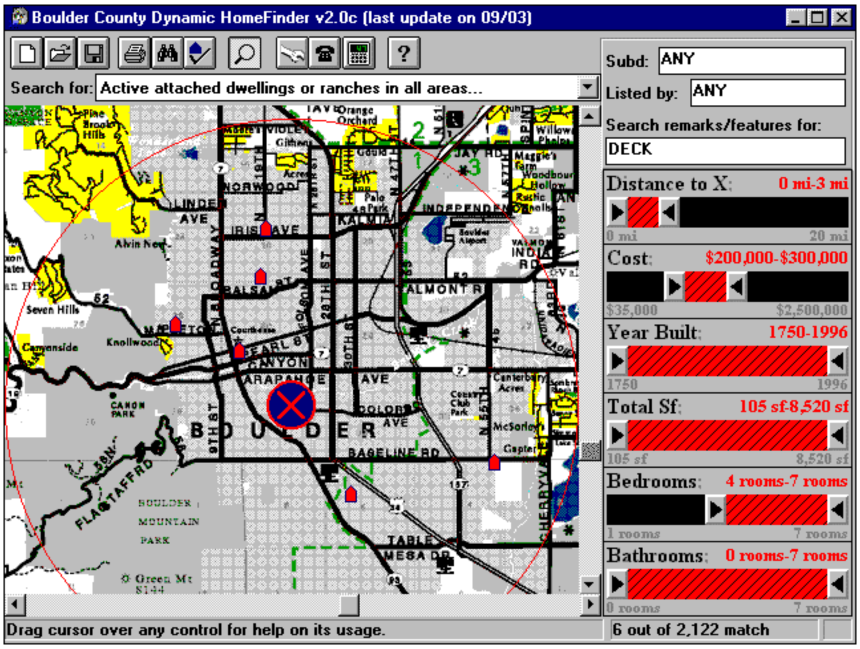
Dynaamisten kyselyiden toimintaa todellisissa informaation etsintäongelmissa on tutkittu useilla testiohjelmilla. Tällaisia testiohjelmia ovat esimerkiksi alkuainetaulukko [AWS92], Dynamic HomeFinder -järjestelmä [WiS92], UNIX-hakemistoselain [LOS92] ja FilmFinder-järjestelmä [AhS94b]. Tässä tutkielmassa esitellään seuraavaksi Dynamic HomeFinder -järjestelmä tarkemmin.

2.2 Dynamic HomeFinder -järjestelmä

Dynamic HomeFinder -järjestelmän lähtökohtana on asunnonetsintä [WiS92]. Asuntoa etsineet tietävät, että asunnonetsintä on työlästä. Sanomalehti-ilmoitukset ja tulostetut listaukset ovat edelleen yleisimmin käytössä olevat menetelmät asunnonetsinnässä. Viimeisen kymmenen vuoden aikana SQL-kielen tapaiset järjestelmät ovat kuitenkin yleistyneet mahdollistamaan asunnon etsinnässä entistä monimutkaisempia kyselyitä.

Kyselykielellä toteutettujen ohjelmistojen ongelmana on se, että ohjelmiston käyttö vaatii opiskelua. Lähes mikään ohjelmisto ei salli asunnonetsijän etsiä asuntoja itse, koska ohjelmiston käyttö on hänelle liian monimutkaista [WiS92]. Ohjelmistoilla on mm. hyvin vaikeaa määritellä sijainteja. Asunnonetsijän täytyy tietää lähiön nimi ja yrittää selvittää onko annettu lähiö paikka, jossa hän haluaa asua. Asunnonetsijä ei yleensä tunne tietokannan sisältöä ja antaa helposti kyselyn, joka on liian yleinen tai rajoittava. Siksi kyselyn tuloksena asunnonetsijä saa joko liian monta vaihtoehtoa tai liian vähän vaihtoehtoja [WiS92].

Dynamic HomeFinder -järjestelmän käyttöliittymä tarjoaa graafisen kuvan sekä kyselyparametreistä että vastaavista tuloksista [WiS92]. Käyttöliittymän vasemmassa osassa on kartta tarkasteltavasta alueesta (kuva 4). Käyttäjä muodostaa kyselyt asettamalla hiiren avulla liukusäätimiin ja painonappeihin halutut parametrien arvot. Kyselyn tulos saadaan yhdistämällä liukusäätimien ja painonappien arvot loogisella operaatiolla AND. Kartalla esitetään keltaisilla pisteillä asunnot, jotka täyttävät kyselyn ehdot.



Kuva 4: Dynamic HomeFinder -järjestelmän käyttöliittymä [WiS92].

Esimerkki 1. Oletetaan, että Matti ja Maija etsivät asuntoa. Dynamic HomeFinder -järjestelmässä kartalle aluksi asetetaan kaksi kiintopistettä, jonka jälkeen kahdella liukusäätimellä määritellään, kuinka kaukana kiintopisteistä asunnon halutaan sijaitsevan.

Matti työskentelee kirjastossa ja omistaa auton. Matti kuitenkin haluaisi, että työmatka ei olisi yli 15 kilometriä. Matti siirtää sijaintimerkin A työpaikkansa kohdalle kartalla ja liukusäätimen A arvon kohtaan 15. Maija työskentelee kaupassa ja haluaisi, että asunto olisi enintään 10 kilometrin päässä työpaikalta, joten Maija siirtää sijaintimerkin B työpaikkansa kohdalle kartalla ja liukusäätimen B arvon kohtaan 10.

Seuraava parametri, joka vaikuttaa kyselyn tulokseen on makuuhuoneiden määrä asunnossa. Koska Matilla ja Maijalla on kaksi lasta, he haluaisivat asunnon, jossa on 2 - 3 makuuhuonetta. He asettavatkin makuuhuoneiden liukusäätimen alkupään arvoksi 2 ja loppupään arvoksi 3.

Seuraavaksi käyttäjä voi määritellä asunnon minimi- ja maksimihinnan. Matti ja Maija ovat päättäneet, että asunto saa maksaa noin 400.000 - 700.000 markkaa. Liukusäätimen, jonka yllä lukee hinta arvoiksi Matti asettaa siis arvot 400k ja 700k. Perhe on päättänyt yrittää löytää omakotitalon. He valitsevat painonapin, jossa lukee talo. Koska Matti omistaa auton, asunnossa tulisi olla autotalli. Matti asettaa painonapin, jossa lukee autotalli päälle. Muut painonapit he asettavat pois päältä.

Nyt on muodostettu kysely:

etsi kaikki ne asunnot, joiden
etäisyys paikasta A $\leq$ 15 kilometriä AND
etäisyys paikasta B $\leq$ 10 kilometriä AND
2 $\leq$ makuuhuoneita $\leq$ 3 AND
400.000 $\leq$ myyntihinta $\leq$ 700.000 AND
talon tyyppi on omakotitalo AND
talossa on autotalli.

Kartalla näkyvät jokaisen kyselyn muutoksen jälkeen ne asunnot, jotka täyttävät silloiset ehdot. Matti ja Maija voivat nyt tutkia kyselyn tuottamia asuntoja tarkemmin valitsemalla jonkin kartalla näkyvistä keltaisista pisteistä. Jos yhtään pistettä ei näy tai pisteitä on liian paljon, on kyselyä helppo muuttaa. □

2.3 Dynaamisten kyselyiden edut

Dynaamisten kyselyiden hyöty on siinä, että ne sallivat käyttäjien tutkia tietokantaa nopeasti, turvallisesti ja jopa leikitellen [Shn94]. Dynaamisillä kyselyillä pystytään kyselyt laatimaan käyttäen liukusäätimiä ja muita käyttöliittymäolioita nopeammin kuin perinteisellä kyselykielellä [WiS92]. Dynaamisen kyselyn tulokset näkyvät koko ajan näyttöruudulla, mikä helpottaa kyselyn hienosäätöä. Siksi aloittelijat oppivat nopeasti laatimaan kyselyjä ja tulkitsemaan tuloksia.

Aloittelevien käyttäjien ei tarvitse muistaa minkäänlaista syntaksia, koska kyselyparametrit esitetaan graafisessa muodossa [WiS92]. Graafiset kyselyparametrit rajoittavat käyttäjän antamia kyselyitä: käyttäjä ei pysty antamaan kyselyn parametreihin sellaisia arvoja, jotka ovat virheellisiä. Siksi ohjelmassa ei enää tarvita virheilmoituksia, joilla ilmoitetaan itse kyselyn muodostuksessa sattuneista virheistä. Lisäksi kaikki käyttäjän tekemät toiminnot ovat peruttavissa.

Aikaisemmat tutkimukset ovat osoittaneet, että dynaamisten kyselyiden käyttöliittymät ovat käyttäjien mielestä helppoja käyttää ja kyselyiden muodostaminen on nopeampaa kuin perinteisillä kyselykielillä [AWS92, LOS92, WiS92]. Tulosten mukaan tietokannan ja kyselyn parametrien visualisointi on kaikkein tärkein osa dynaamisia kyselyitä.

2.4 Dynaamisten kyselyiden ongelmat

Dynaamiset kyselyt ovat lupaavia, mutta ne ovat kaukana täydellisyydestä. Tärkein ongelma on se, että dynaamisten kyselyiden käyttöliittymän ohjelmointi on sovelluskohtaista [WiS92]. Yhdenkin uuden atribuutin lisääminen kyselyparametreihin vaatii muutoksia ohjelmakoodiin. Ongelman ratkaisemiseksi on esitetty ainakin ohjelmistoa nimeltä IVEE (Information Visualization & Exploration Environment) [AhW95, Wis95]. IVEE osaa lukea tietokannan relaation ja luoda sen mukaisen graafisen käyttöliittymän, joka sisältää tarvittavat käyttöliittymäoliot ja tiedon visualisointitavat.

Dynaamisissa kyselyissä tieto täytyy järjestää tietyllä tavalla ja on käytettävä tietorakenteita, joista tiedon hakeminen on tehokkaampaa. [WiS92]. Tälläisen tietorakenteen löytäminen ei aina ole helppoa ja jokainen attribuutti saattaa vaatia erilaisen tietorakenteen tehokkaan etsinnän varmistamiseksi. Jain ja Shneiderman [JaS93] ovat vertailleet erilaisten tietorakenteiden käyttöä dynaamisten kyselyiden toteutukseen.

Näytön rajallinen koko tekee toimintojen ja tulosten esittämisen vaikeaksi yhdellä ruudulla [WiS92]. Visuaalisen yleiskuvan kannalta olisi tärkeää saada molemmat mahtumaan yhdelle ruudulle. Dynaamisissa kyselyissa visualisoinnin tulisi olla graafinen, sillä dynaamiset kyselyt toimivat parhaiten tiedossa, joka on helppo esittää graafisesti.

Jotta dynaamisten kyselyiden käyttöliittymä onnistuisi, täytyy useita asioita harkita tarkoin. Kyselyille täytyy löytää hyvä visualisointikeino, kuten kartta tai taulukko. Kyselyn hallintaosa täytyy sijoittaa loogisella tavalla, jotta silmän ja hiiren liikuttelu olisi minimaalista. Liukusäätimet ja muut kyselyn hallintaparametrit täytyy toteuttaa niin, että aloittelijoidenkin on helppo käyttää niitä [WiS92].

3 Suodatintoiminnot

3.1 Suodatintoimintojen peruskäsitteitä

Dynaamisen kyselyn *suodatin* (dynamic query filter) on kyselyn parametri, jota muutetaan liukusäädintä, painonappia tai muuta vastaavaa käyttöliittymäoliota käyttäen [AhS94b]. Esimerkiksi Dynamic HomeFinder -järjestelmä [WiS92] antaa käyttäjälle mahdollisuuden säätää asuntojen hintojen ylä- ja alarajaa. Jokainen tälläinen kyselyn komponentti toimii kuten suodatin vähentäen tulosjoukon kokoa. Parametrit yhdistetään loogisella operaatiolla AND. Jos kyselyssä tarvitaan loogista operaatiota OR, joudutaan kysely muodostamaan peräkkäisistä kyselyistä.

Toinen suodatintoimintojen käyttöliittymän osa on se, kuinka tulosjoukko esitetään ruudulla. *Tähtikenttäesitys* (starfield display) esittää tulosjoukon jatkuvasti ja mahdollistaa tuhansien olioiden esittämisen [AhS94b]. Tähtikentässä tulokset voidaan esittää esimerkiksi *hajautuspisteinä* (scatterplot). Hajautuspiste on ruudulla oleva neliömäinen piste, jonka sijainti määräytyy pisteen kuvaaman olion attribuuteista kaksiulotteisessa avaruudessa. Esimerkiksi FilmFinder -järjestelmässä [AhS94b] on käytetty tähtikenttäesitystä, jossa x-koordinaatti on elokuvan valmistusvuosi ja y-koordinaatti on elokuvan pituus minuuteissa. Tähtikenttäesitys mahdollistaa lisäksi tulosten valinnan ja suurentamisen.

Tässä tutkielmassa keskitytään suodatintoimintoihin, jotka on toteutettu käyttäen ns. *Magic lens -suodattimia* (Magic lens filters). Magic lens -suodattimet ovat käyttöliittymätyökaluja, joilla yhdistetään mielivaltaisen muotoinen alue operaattorilla [BSP+93, BSF+94, SFB94, FiS95]. Tälläisia operattoreita ovat esimerkiksi loogiset operaatiot AND, OR ja NOT.

Operaattori muuttaa *näkymää* (lenses) olioihin, jotka ovat näkymän alueella [FiS95]. Operaattori voi olla melko yleinen. Operaattorilla voidaan käsitellä allaolevien olioiden tietorakenteita ja uudelleenmuotoilla olion tietoja muodostamaan uusi näkymä.

Magic lens -suodattimet ovat *tilallisesti sidottu* (spatially bounded) [FiS95]. Ne voidaan parametrisoida ja asettaa interaktiivisesti kuvaruudulla olevien ohjelmien päälle. Suodattimet voivat olla osittain päällekkäin, jolloin ne muodostavat yhdistettyjä kyselyitä, joita voidaan kontrolloida.

3.2 Suodatintoimintojen ryhmittely

Suodatintoiminnot voidaan ryhmitellä usealla tavalla. Toteutuksen kannalta suodatintoiminnot jaetaan *kiinteisiin suodattimiin* ja *siirrettäviin suodattimiin*. Kiinteällä suodattimella tarkoitetaan suodatinta, jonka paikka ruudulla on kiinteä ja jonka operaattorin toiminta kohdistuu koko tietokantaan. Jos sovelluksessa on useita kiinteitä suodattimia, suodattimien operaattorit yhdistetään käyttäen loogista operaatiota AND tai käyttäjän määrittelemillä operaatioilla. Esimerkiksi Dynamic HomeFinder -järjestelmän [WiS92] toteutuksessa on käytetty kiinteitä suodatintoimintoja.

Siirrettävällä suodattimella tarkoitetaan suodatinta, jota voidaan siirtää sovelluksen alueella [SFB94]. Siirrettävän suodattimen operaattori voidaan kohdistaa halutessa vain osaan tietokantaa. Siirrettävän suodattimen operaattoria ja vaikutusaluetta voidaan muuttaa tarvittaessa. Jos halutaan muodostaa yhdistettyjä operaatioita, siirretään useampia siirrettäviä suodattimia osittain tai kokonaan toistensa päälle.

4 Siirrettävät suodatintoiminnot

4.1 Totuusarvoiset kyselyt

Joukkoa näkymiä voidaan käyttää luomaan dynaamisia kyselyitä tietoon, jos tieto voidaan esittää hajautuspisteinä [FiS95]. Tälläiset näkymät voidaan toteuttaa käyttäen Magic lens -suodattimia. Eräs tapa toteuttaa Magic lens -suodattimet on käyttää ikkunoita.

Jokainen näkymä toimii suodattimena, joka käsittelee tiettyä tiedon attribuuttia [FiS95]. Liukusäädinsuodatin kontrolloi numeerisen tiedon rajoja. Painonapit ja muut kontrollivälineet voivat kontrolloida näkymissä muita toimintoja. Kun näkymät ovat kokonaan tai osittain toistensa päällä, niiden toiminnot yhdistetään. Tämä tarjoaa helpon tavan muodostaa monimutkaisia kyselyitä. Jokainen näkymä määrittelee suodatusfunktion ja *yhdistämismuodon* (composition mode), jolla määritellään kuinka suodatinfunktion tulos yhdistetään alla olevien näkymien tuloksiin [FiS95].

Jotta Magic lens -suodattimilla voitaisiin muodostaa totuusarvoisia kyselyitä, täytyy määritellä kuinka suodattimet yhdistetään [FiS95]. Täydellisen totuusarvoisen toimivuuden tarjoamiseksi on määriteltävä mekanismit loogisiin operaatioihin AND, OR, NOT sekä eri operaatioiden yhdistämiseen tai ryhmittelyyn.

Määritelmä 1: Olkoon näkymä L=(F,M), missä F on suodatin ja M on totuusarvoinen operaattori. Suodatin F määrittelee tuloksen laskentatavan suodattimen jollekin tiedolle. Operaattori M määrittelee, kuinka tulos yhdistetään alla olevien suodattimien tuloksiin [FiS95].

Esimerkki 2: Olkoon annettuna näkymät L1=(F1,OR) ja L2=(F2,AND). Jos näkymä L1 asetetaan näkymän L2 päälle, muodostuu lause (F1 OR F2). Toisaalta, jos näkymä L2 asetetaan näkymän L1 päälle muodostuu lause (F2 AND F1). □

Looginen operaatio NOT voidaan määritellä näkymänä, jonka suodatin kääntää sisääntulevan tiedon merkityksen [FiS95]. Jos merkitään puuttuvaa suodatinta merkinnällä NULL, voidaan samaa formaalia määritelmää käyttäen määritellä päinvastainen näkymä N = (NULL,NOT). Tällöin näkymä N suorittaa loogisen operaation NOT kaikille allaolevien näkymien tuloksille.

Esimerkki 3: Olkoon kysely (F1 OR NOT F2), missä F1 ja F2 ovat suodattimia eri attribuuteille. Toteuttaakseen tämän kyselyn käyttäjä asettaa aluksi suodattimen F2. Tämän jälkeen käyttäjä asettaa suodattimen NOT N suodattimen F2 päälle. Lopuksi käyttäjä asettaa suodattimen F1 operaattorilla OR edellisten päälle. □

Näkymien yhdistämiseksi täytyy määritellä mekanismi kootulle lauseelle, joka on määritelty joukolla näkymiä [FiS95]. Tähän päästään tarjoamalla operaatio, joka korvaa joukon näkymiä yhdellä yhdistetyllä näkymällä, joka on semanttisesti ekvivalentti.

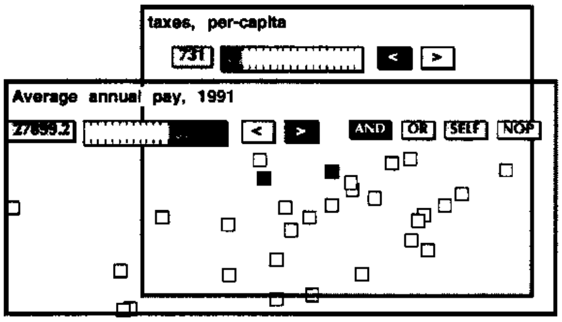
Käyttäjä luo yhdistetyn näkymän käyttäen *läpipainonappia* (click-through button) [BSF+94]. Kaikista läpipainonapin alla olevista näkymistä luodaan yhdistetty näkymä, joka muodostaa saman kyselyn kuin joukko näkymiä. Tuloksena olevalla näkymällä on yhdistämisoperaattori, joka määrittelee, kuinka yhdistetty näkymä yhdistetään muihin näkymiin. Käyttäjä voi käsitellä uutta yhdistettyä näkymää samalla tavalla kuin muita näkymiä. Yhdistetty näkymä tarjoaa helpon tavan yhdistää monimutkaiset kyselyt yhdeksi kokonaisuudeksi. Yhdistetyt näkymät voivat sisältää muita yhdistettyjä näkymiä. Näin voidaan muodostaa mielivaltaisen monimutkaisia kyselyitä.

Esimerkki 4: Olkoon kysely (F1 AND F2) OR (F3 AND F4). Kyselyn muodostamiseksi luodaan yhdistetyt näkymät suluissa oleville lauseille. Olkoon siis C1 = (F1 AND F2) ja C2 = (F3 AND F4). Määrittelemällä yhdistetyn suodattimen operaattoriksi OR luodaan haluttu kysely asettamalla näkymä C1 näkymän C2 päälle. □

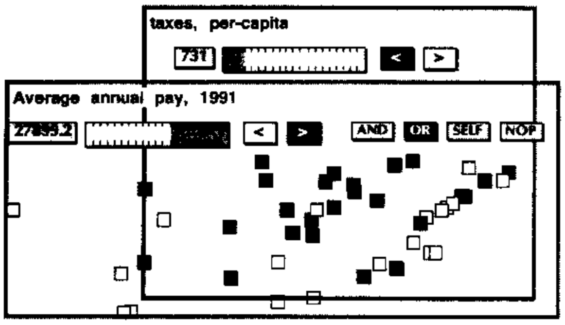
Loogisilla operaatioilla AND, OR, NOT ja yhdistämisellä voidaan esittää mikä totuusarvoinen kysely tahansa [FiS95]. Monimutkaiset kyselyt voidaan muodostaa suorakäsittelemällä näkymiä ja niiden operaattoreita. Hyödylliset kyselyt voidaan tallettaa yhdistettyinä näkyminä. Tulosnäkymiä voidaan käyttää muodostettaessa vielä monimutkaisempia kyselyitä. Mallilla pystytään toteuttamaan kaikki loogiset operaatiot, vaikka vain kolme yleisintä loogista operaatiota on toteutettu [FiS95].

Esimerkki 5: Edellä kuvatun idean esittelemiseksi on toteutettu ohjelma, jolla voidaan tutkia Yhdysvaltojen väestonlaskentatietokannan tietoja [FiS95]. Toteutuksessa jokainen näkymä toteutettiin X -ikkunointijärjestelmän ikkunana. Jokainen kaupunki esitetään sinireunaisena valkoisena neliönä kohdassa, joka vastaa kaupungin sijaintia. Punaisina neliöinä esitetyt kaupungit läpäisevät kyselyn. Suodatinnäkymään vaikuttava tietoattribuutti esitetään ikkunan nimiosassa. Näkymän vasemmassa yläreunassa on liukusäädin, jolla kontrolloidaan kyselyn attribuutin arvoja ja kenttä, jossa on liukusäätimellä valittu arvo. Liukusäätimen vieressä on kaksi painonappia, joilla voidaan muuttaa vertailuoperaatiota. Näkymän oikeassa yläreunassa olevat painonapit ovat näkyvissä, jos on käytetty yhdistämisoperaatiota. Ikkunan oikeassa reunassa olevia painonappeja käytetään luomaan uusia näkymiä.
□

Näkymän yhdistämisoperaattori määritellään *valintanappien* (radio button) avulla [FiS95]. Valintanapit ovat painonappeja, joista vain yksi voi olla valittuna kerrallaan. Painonapeilla AND ja OR asetetaan yhdistämisoperaattori. Painonappilla SELF saadaan aikaan toiminto, jossa näkymän tulos muodostetaan käyttäen näkymän omaa suodatinta ohittaen alla olevien näkymien tulokset. Painonapilla NOP asetetaan suodatinoperaattori NULL:ksi, jolloin suodatin ei suorita mitään operaatiota alla olevalle tiedolle. Esimerkiksi kuvassa 6 on muodostettu kysely käyttäen yhdistämisoperaattoria AND ja kuvassa 7 käyttäen yhdistämisoperaattoria OR.

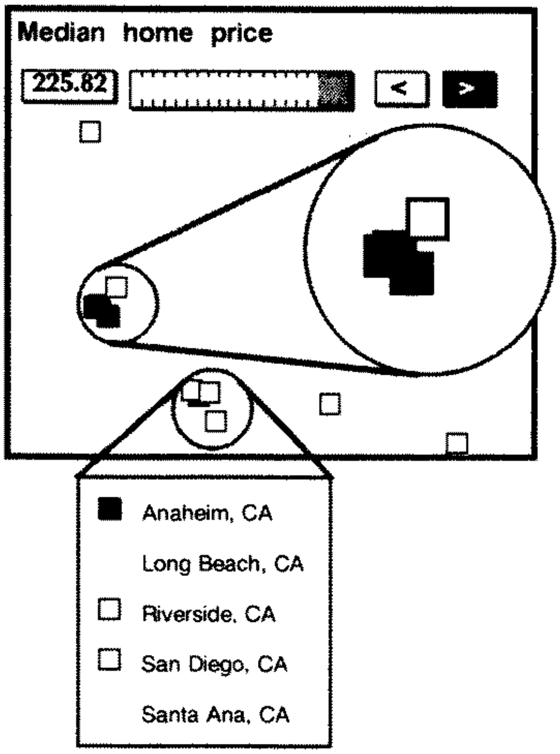


Kuva 5: Korkeat palkat AND alhaiset verot [FiS95].



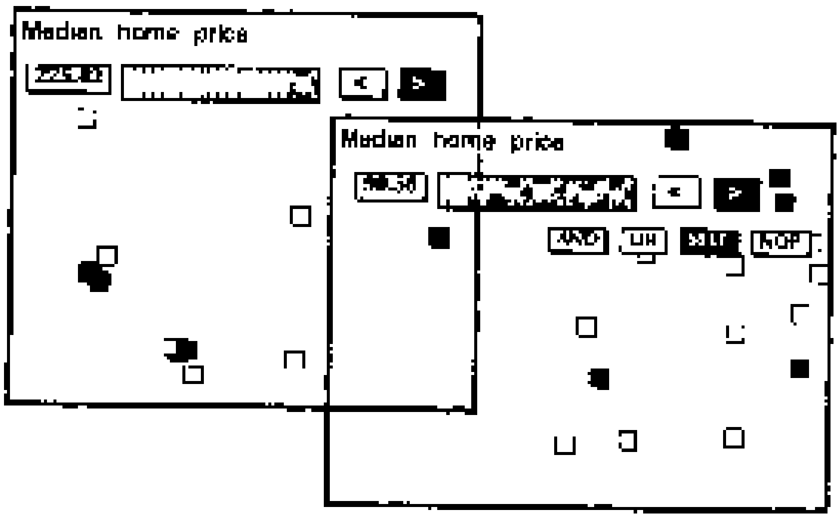
Kuva 6: Korkeat palkat OR alhaiset verot [FiS95].

Hajautuspiste-esityksessä tiedossa on usein *ryppäitä* (clumps), joissa useampia tietoalkioita on näytöllä lähekkäin tai jopa osittain päällekkäin [FiS95]. Kuvassa 8 on esimerkki ryppäistä ja useammasta suodattimesta, jotka auttavat tiedon esittämistä. Toiseen ryppääseen on asetettu *suurennusnäkymä* (magnifying lens), jotta ryppään eri pisteet näkyisivät. Toiseen ryppääseen on asetettu *esiintuontinäkymä* (call-out lens), joka esittää ryppään pisteet ikkunassa listana. Tälläinen lista helpottaa olioiden tunnistamista. Ikkunan listassa esitetään olioiden tunnisteet ja tunnisteiden edessä hajautuspiste. Listassa näkyvät hajautuspisteet ovat aktiivisia. Niiden päälle asetetut suodattimet toimivat samoin kuin muut hajautuspisteet, jos niiden päälle asetetaan suodattimia.



Kuva 7: Suurennusnäkymä ja esilletuontinäkymä [FiS95].

Useita kyselyitä voidaan toteuttaa samanaikaisesti asettamalla erilaisia näkymiä näytön eri osiin [FiS95]. Jokainen näkymä ja jokainen kysely voidaan erikseen muotoilla muuttamalla näkymän kontrolleja (kuva 8).



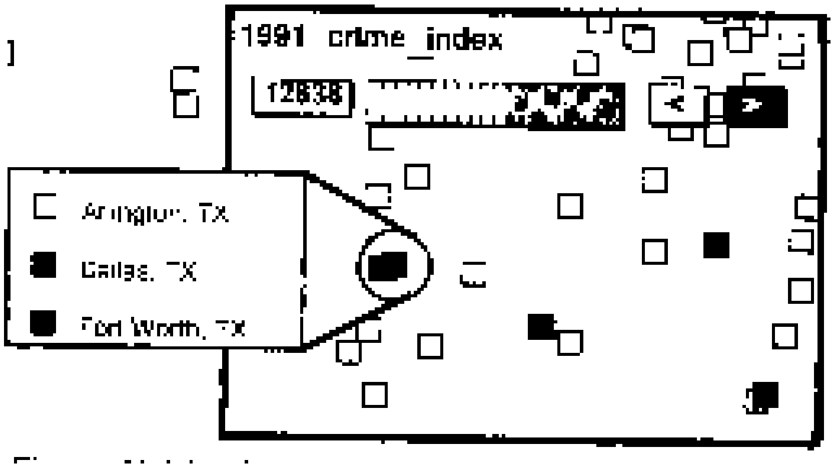
Kuva 8: Edulliset asunnot Californiassa ja Texasissa [FiS95].

4.2 Reaaliarvoiset kyselyt

Tehokkaampien kyselyiden tukemiseksi ohjelmistoa laajennettiin siten, että suodattimet liittivät attribuuttiin *reaaliarvoisen* (real-valued) pistearvon väliltä [0,1] [FiS95]. Attribuutti, jonka pistearvo on 0 epäonnistuu suodatuksessa kokonaan ja attribuutti pistearvolla 1 onnistuu täydellisesti. Välille jääneet attribuuttien pistearvot onnistuvat suodatuksessa osittain.

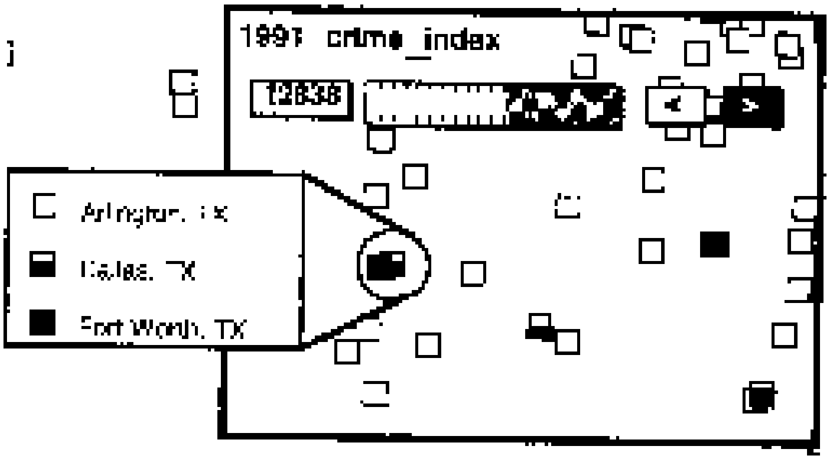
Pistearvot esitetään esimerkin 5 ohjelmistossa visuaalisesti jokainen alkio osittain täytettynä neliönä. Mitä korkeampi pistearvo sitä enemmän neliö on täytetty [FiS95].

Esimerkki 6: Totuusarvoisella kyselyllä haetaan kaupungit, joiden rikosindeksi ylittää arvon 12838 (kuva 9). Kuvassa 10 on esiintuontinäkymää käyttäen valittu osa tietokantaa tarkempaan tarkasteluun.



Kuva 9: Totuusarvoinen kysely [FiS95].

Käyttäen reaaliarvoisia kyselyitä näemme valittujen olioiden realiarvoisten pistearvojen erot (Kuva 10). Esitetyt tulokset lasketaan seuraavasti. Koska vertailussa käytetään operaattoria suurempi kuin, asetetaan kaikkien liukusäätimen valittua arvoa pienempien attribuuttien arvojen pistearvoksi 0. Kuvan 11 esimerkissä tälläinen kaupunki on Arlington. Suurimman tietokannassa olevan attribuutin arvon pistearvoksi asetetaan 1. Kaikki muut kyselyn täyttävien olioiden pistearvot saadaan jakamalla attribuutin arvo maksimiarvolla.

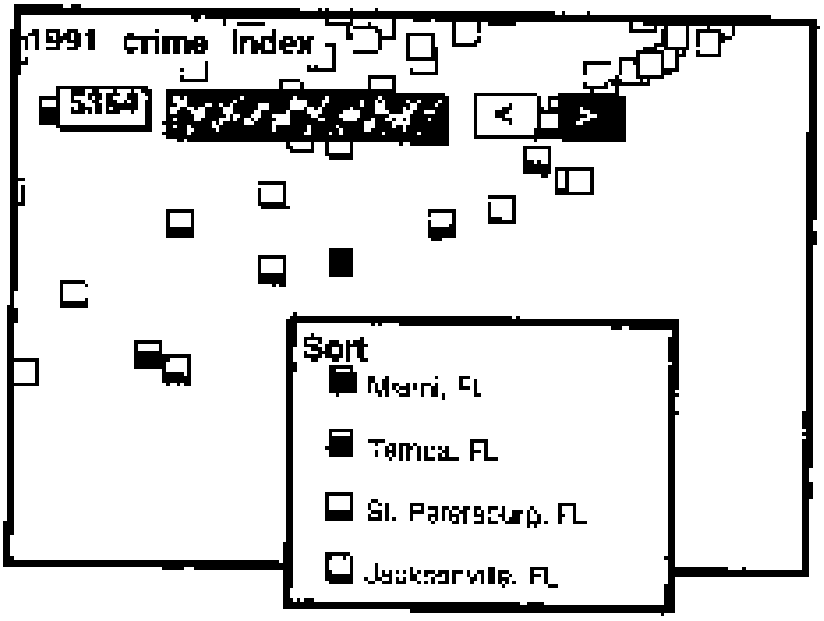


Kuva 10: Vastaava realiarvoinen kysely [FiS95].

Reaaliarvoiset kyselyt vaativat reaaliarvoiset yhdistämisoperaattorit [FiS95]. Reaaliarvoiset yhdistämisoperaattorit MAX ja MIN vastaavat loogisia operaattoreita AND ja OR. Eli MAX ja MIN toimivat samoin arvoille 0.0 ja 1.0 kuin AND ja OR arvoille 0 (epätosi) ja 1 (tosi). Reaaliarvoinen suodatin NOT palauttaa 1.0 vähennettynä reaaliarvoisesta pistearvosta.

Loogiset operaattorit voidaan yhdistaa mielivaltaisen monen loogisen operaattorin kanssa. Myös reaaliarvoiset kyselyt voidaan yhdistää mielivaltaisen monen numeerisen operaattorin avulla [FiS95]. Tälläisiä operaattoreita voivat olla tilastolliset muunnokset (sqrt, arcsin jne.), joita käyttäen muutetaan tiedon jakautumista.

Reaaliarvoisten suodattimien esityksessä voidaan käyttää laskentaa [FiS95]. Tällaista laskentaa on esimerkiksi attribuuttien arvojen lajittelu ja suurimman attribuutin arvon etsintä. Esimerkiksi kuvassa 12 on *lajittelunäkymä* (sorting lens), joka on asetettu realiarvoisen suodattimen päälle. Lajittelunäkymä lajittelee näkymän alueella olevat attribuuttien reaaliarvoiset pistearvot ja esittää ne ikkunassa listana. Kuvassa 12 on lajittelunäkymällä lajiteltu rikosindeksin pistearvojen mukaisesti näkymän alueella olevat kyselyn ehdot täyttävät kaupungit.



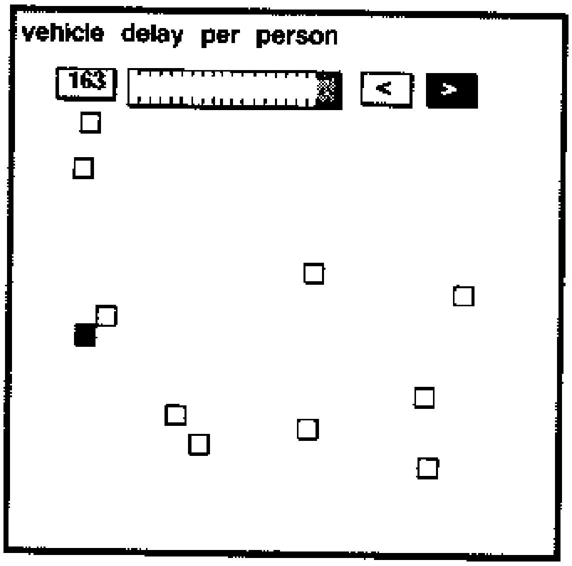
Kuva 11: Lajittelunäkymä [FiS95].

4.3 Puuttuvan tiedon esitys

Tietokannoissa on yleisesti tietoa, josta puuttuu joidenkin attribuuttien arvot [FiS95]. Täydellisen tiedonesityssysteemin täytyy esittää myös puuttuva tieto. Tiedon esittämisessä on kaksi ongelmaa: kuinka tieto esitetään visuaalisesti ja kuinka puuttuvaan tietoon kohdistuvat kyselyt toteutetaan. Esimerkissä 5 kuvatussa ohjelmistossa puuttuva tieto esitetään, kun käytetään erityistä näkymää. Jos kaupungilta puuttuu kyselyn attribuutin arvo, sitä ei normaalisti esitetä ruudulla (kuva 12).

Puuttuvan tiedon näkymässä (show missing data) oliot, joista attribuutin arvo puuttuu, esitetään hajautuspisteestä poikkeavalla tavalla [FiS95]. Puuttuvan tiedon näkymän avulla käyttäjä pystyy näkemään, kuinka monelta ilmentymältä kyselyn attribuutin arvo puuttuu. Esimerkissä 5 kuvatussa ohjelmistossa kaupungit, joista kyselyn attribuutin arvo puuttuu esitetään näytöllä X merkillä (kuva 14).

Määritelmä 2: Merkitään puuttuvaa tietoa arvolla NULL. Tällöin yhdistysoperaattoreille voidaan määritellä, kuinka ne toimivat puuttuvan tiedon tapauksissa. Eri operaatiot tulkitaan



Kuva 12: Totuusarvoinen suodatin [FiS95].

OR(a,NULL) = a

MAX(a,NULL) = a

F(a,NULL) = NULL kaikille muille yhdistysoperaattoreille F.

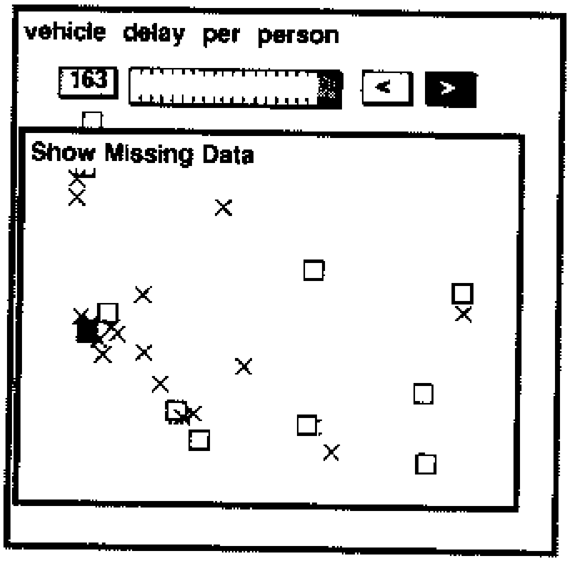
Missä a on toisen vertailtavan attribuutin arvo ja muita yhdistysoperaattoreita F ovat esimerkiksi AND ja MIN. □

Puuttuvaa tietoa sisältävä olio voi ilmaantua uudestaan, jos toinen suodatin asetetaan sen päälle [FiS95]. Tälläisiä suodattimia ovat MAX ja OR.

5 Suodatintoimintojen arviointia

5.1 Edut

Kiinteiden suodatintoimintojen etuja on jo käsitelty dynaamisten kyselyiden arvioinnin yhteydessä. Dynaamisten kyselyiden edut pätevät myös siirrettävissä suodatintoiminnoissa. Siirrettävillä suodatintoiminnoilla pystytään luomaan yleisiä totuusarvoisia kyselyitä



Kuva 13: Puuttuvan tiedon suodattin [FiS95].

suorakäsittelyä käyttäen. Siirrettävät suodatintoiminnot tarjoavat useita samanaikaisia näkymiä ja yhtenäisen tavan esittää vaihtoehtoisia näkymiä tietoon [FiS95].

Siirrettävillä suodatintoiminnoilla laajennetut dynaamiset kyselyt ratkaisevat joitakin aikaisemmin mainittuja dynaamisten kyselyiden ongelmia [FiS95]. Yksi tälläinen ongelma oli se, että dynaamisilla kyselyillä pystyttiin esittämään vain yksinkertaisia kyselyitä [WiS92]. Siirrettävillä suodatintoiminnoilla laajennettuna dynaamisilla kyselyillä pystytään esittämään mielivaltaisen monimutkaisia kyselyitä [FiS95].

Siirrettävillä suodatintoiminnoilla laajennetut dynaamiset kyselyt pystytään myös soveltamaan vain haluttuun osaan tietokantaa [FiS95]. Näin pystytään soveltamaan erilaisia kyselyitä eri tietokannan osiin ja saadaan esille tietokannan eri osien erot.

Laajentamalla suodatintoimintoja reaaliarvoisilla näkymillä pystytään kyselyn tuloksia vertailemaan tarkemmin [FiS95]. Reaaliarvoisien näkymien avulla käyttäjä pystyy lajittelemaan tulokset ja näkemään tulosten erot selkeästi.

5.2 Ongelmat

Kiinteiden suodatintoimintojen ongelmat ovat samoja kuin dynaamisisten kyselyiden arvioinnissa esitetyt. Siirrettävät suodatintoiminnot lisäävät ohjelmiston toteutuksen ongelmia.

Siirrettävien suodatintoimintojen toteutus täytyy laajentaa soveltumaan myös muunlaisiin visualisointitapoihin kuin hajautuspisteisiin, sillä hajautuspiste-esityksen käyttö vaatii jotain tapaa esittää tieto kaksiulotteisessa avaruudessa [FiS95]. Kaksiulotteisen esityksen löytäminen ei aina ole mahdollista.

Siirrettävien suodatintoimintojen kyselyn hallinta tulee laajentaa soveltumaan muihinkin käyttöliittymäolioihin kuin liukusäätimeen. Sillä liukusäädin vaatii numeerisen attribuutin arvon [FiS95]. Tälläisia käyttöliittymäolioita ovat esimerkiksi painonappi ja aakkosnumeerinen liukusäädin.

6 Yhteenveto

Dynaamiset kyselyt tarjoavat eloisan uuden suunnan tietokantakyselyihin. Useat ongelmat, jotka ovat vaikeita hallita hakusanaorientoitunutta komentokieltä käyttäen ovat helposti käsiteltäviä dynaamisillä kyselyillä. Tutkimuksissa toteutetut testijärjestelmät osoittavat, että dynaamiset kyselyt sopivat hyvin tiedonhakuun, joissa tieto voidaan esittää graafisesti. Dynaamisista kyselyistä saattaa tulla yleinen lähestymistapa, joka lisätään jokaiseen tietokantajärjestelmään, taulukkolaskentaohjelmaan ja moniin muihin sovelluksiin.

Siirrettävien suodatintoimintojen käyttö tiedon etsintään ja visualisointiin luo käyttäjälle mallin, joka on ilmaisuvoimainen, helppokäyttöinen ja ymmärrettävä. Koko totuusarvoinen kyselykieli voidaan esittää graafisesti tarjoamalla yhdistämisoperaatiot ja yhdistetyt lauseet. Kieltä voidaan myös laajentaa reaaliarvoisilla funktioilla, jotka selkeästi käsittelevät puuttuvat attribuuttien arvot. Tehokkaita kyselyitä voidaan toteuttaa vaiheittain ja tallettaa myöhempää käyttöä varten. Siirrettävät suodatintoiminnot tarjoavat yhtenäisen, tehokkaan ja laajennettavan tavan tiedon visualisointiin ja etsintään ohjelmistojärjestelmissä.

Viitteet

[AhS94a] Ahlberg, C., Shneiderman, B., The AlphaSlider: A Compact and Rapid Selector. Proc. ACM Conf. Human Factors in Computing Systems, Boston, Massachusetts, USA, April 24–28, 1994, 365–371.

[AhS94b] Ahlberg, C., Shneiderman, B., Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays. Proc. ACM Conf. Human Factors in Computing Systems, Boston, Massachusetts, USA, April 24–28, 1994, 313–317.

[AhW95] Ahlberg, C., Wistrand, E., IVEE: An Environment for Automatic Creation of Dynamic Queries Applications. Technical Report SSKKII-95.03, SSKKII, Göteborg University, Sweden, 1995.

[AWS92] Ahlberg, C., Williamson, C., Shneiderman, B., Dynamic Queries for Information Exploration: An Implementation and Evaluation. Proc. ACM Conf. Human Factors in Computing Systems, Monterey, California, USA, May 3–7, 1992, 619–626.

[BSF+94] Bier, E. A., Stone, M. C., Fishkin, K., Buxton, W., Baudel, T., A Taxonomy of See-Through Tools. Proc. ACM Conf. Human Factors in Computing Systems, Boston, Massachusetts, USA, April 24–28, 1994, 358–364.

[BSP+93] Bier, E. A., Stone, M. C., Fishkin, K., Buxton, W., DeRose, T. D., Toolglass and Magic Lenses: The See-Through Interface. Proc. ACM Conf. Computer Graphics, Anaheim, California, USA, August 1–6, 1993, 73–80.

[FiS95] Fishkin, K., Stone, M. C., Enhanced Dynamic Queries via Movable Filters. Proc. ACM Conf. Human Factors in Computing Systems, Denver, Colorado, USA, May 7–11, 1995, 415–420.

[JaS93] Jain, V., Shneiderman, B., Data Structures for Dynamic Queries: An Analytical and Experimental Evaluation. Technical Report CS-TR-3133, University of Maryland, 1993.

[KKS88] Kim, H., Korth, H., Silberschatz, A., PICASSO: A Graphical Query Language. Software - Practice and Experience, 18,3 (1988),169–203.

[LOS92] Liao, H. S., Osada, M., Shneiderman, B., Browsing Unix Directories With Dynamic Queries: An Evaluation of Tree Information Display Tecniques. Technical Report CS-TR-2841, University of Maryland, 1992.

[OLS93] Osada, M., Liao, H. S., Shneiderman, B., AlphaSlider: Shearching Textual Lists with Sliders. Technical Report CS-TR-3079, University of Maryland, 1993.

[SFB94] Stone, M. C., Fishkin, K., Bier, E. A., The Movable Filter as a User Interface Tool. Proc. ACM Conf. Human Factors in Computing Systems, Boston, Massachusetts, USA, April 24–28, 1994, 306–312.

[Shn83] Shneiderman, B., Direct Manipulation: A Step Beyond Programming Languages. Technical Report CS-TR-1245, University of Maryland, 1983.

[Shn94] Shneiderman, B., Dynamic Queries for Visual Information Seeking. Technical Report CS-TR-3022, University of Maryland, 1994.

[Wis95] Wistrand, E., Visualization Methods for Dynamic Queries Databases. Technical Report SSKKII-95.01, SSKKII, Göteborg University, Sweden, 1995.

[WiS92] Williamson, C., Shneiderman, B., The Dynamic HomeFinder: Evaluating Dynamic Queries in a Real-Estate Information Exploration. Technical Report CS-TR-2819, University of Maryland, 1992.

[Zlo75] Zloof, E., Query by Example. Proc. AFIPS Conf. National Computer Conference, Anaheim, California, USA, May 19–22, 1975, 431–437.