KEDA - Kemistin data-analysaattori
Loppuraportti

Taisto Gerdt
Timo Kääriäinen
Mika Miettinen
Pasi Rastas
Jari Tuominiemi

Ohjelmistotuotantoprojekti
Helsingin yliopisto
Tietojenkäsittelytieteen laitos
5. joulukuuta 2002

Versiohistoria:

Päiväys	Muutokset	Muutosten tekijä
26.11.2002	Alustava versio	Pasi Rastas
3.12.2002	Ryhmätapaamisessa 26.11 esitetyt muutokset	Pasi Rastas
5.12.2002	Ryhmätapaamisessa 3.12 esitetyt muutokset	Pasi Rastas

Sisältö

1 Johdanto

Tällä dokumentilla lopetetaan ohjelmistotuotantoprojektiryhmä KEDA:n tuotantoprosessi. Dokumentin tarkoitus on koota yhteenveto ryhmän aikaansaannoksista, kokemuksista ja tehdyn työn määrästä. Tämän dokumentin tarkoitus on myös tukea mahdollista ohjelmiston jatkokehittelyä. KEDA-ryhmä sai projektin projektisuunnitelmassa kuvatulla tavalla päätökseen, ja tuotettu ohjelmisto täytti hyvin vaatimusmäärittelyn vaatimukset.

Ryhmän jäsenet olivat Taisto Gerdt, Timo Kääriäinen, Mika Miettinen, Pasi Rastas ja Jari Tuominiemi. Ryhmän ohjaajana toimi Katja Saarela(ent. Nyman) ja vastuuhenkilönä toimi Markus Ekholm. Ryhmän asiakkaita olivat Raimo Ketola(VTT), Juho Rousu(TKTL) ja Ari Rantanen(TKTL).

2 Työn kulku

Ryhmän tavoite oli tuottaa ohjelmisto molekyylin positionaalisten isotopomeerien jakauman laskemiseksi massaspektrometrin tuottamasta datasta. Tähän käytetty laskentamenetelmä [Ran02] on kehitetty vasta hiljan, joten työn luonne oli hyvin kokeellinen. Tämän takia työssä päätettiin käyttää spiraalimallia, mikä osoittautui hyväksi valinnaksi. Projektin aikana ehdimme tehdä kaksi spiraalikierrosta. Spiraalikierrosten välissä järjestimme demotilaisuuden, jossa saimme tärkeää palautetta asiakkaalta toiselle spiraalikierrokselle.

Ohjelmisto toteutettiin MATLAB ohjelmointikielellä. Useimmille projektin työntekijöille MATLAB oli aivan outo, joten projektin alussa MATLAB:in opetteluun meni jonkin verran aikaa. Projekti pysyi todella hyvin aikataulussa. Itseasiassa olimme hieman edellä suunnitellusta toisen iteraation toteutuksen alkaessa. Ylimääräinen aika päätettiin käyttää kattavampaan testaukseen, sillä ensimmäisen iteraation jälkeen testaus oli vielä hieman vajavaista. Testaus oli tässä projektissa vaikeaa, sillä asiakkaalla ei ollut mitattuna syötteitä laskentamentelmän uusille ominaisuuksille. Syötetiedostoja ei muutenkaan ollut kuin muutama. Päätimme tyytyä testaamaan lähinnä yksittäisten komponenttien toimintaa. Projektin aikataulu on esitetty projektisuunnitelmassa.

Projektisuunnitelmassa arvoiduista riskeistä laitteisto-ongelmat aiheuttivat hieman vaikeuksia pariin otteeseen laitoksen fs-koneen ollessa kaatuneena. Nämä kaatumiset tapahtuivat onneksi sellaisina ajankohtina, etteivät ne vaikuttaneet projektin aikataulussa pysymiseen tai edistymiseen. Esimerkiksi fs-koneen kaatuminen projektin demopäivänä 17.10 olisi voinut merkitä demotilaisuuden peruuntumista. Myöskään ohjelmointivaikeudet eivät vaikuttaneet projektiin, vaan MATLAB-kielen sisäistäminen tapahtui lähes ongelmitta. Pieni ongelma oli asiakkaan määrittämä vaatimus, että ohjelmiston pitäisi toimia Windows ympäristössä. Tämä aiheutti ongelmia, koska laitoksen koneista löytyi vain Linux-koneita MATLAB:illa varustettuna. Ongelman ratkaisi kuitenkin Timo, jolla oli käytettävissä opiskelijaversio MATLAB-ohjelmistosta. Työnjako olisi myös voinut olla tasaisempi, sillä Mika teki noin 15% enemmän työtunteja kuin muut keskimäärin.

3 Dokumentit

Projektin dokumentointia olisi voinut keventää, ainakin ensimmäisen iteraation osalta. Esimerkiksi dokumenttien jättäminen avoimiksi ensimmäisellä iteraatiolla olisi voinut olla hyvä idea. Tällöin Mikan tekemä työmääräkin olisi ollut muiden tekemien työmäärien tasolla. Esimerkiksi ohjelmistotuotantoprojektin kotisivulla olisi voinut ollut olla yleisiä ohjeita dokumentoinnin toteuttamisesta spiraalimallisessa projektissa.

3.1 Ensimmäinen iteraatio

Projektisuunnitelma

Projektisuunnitelmassa määritellyt aikataulut ja muut suunnitelmat pitivät hyvin paikkaansa. Osa projektisuunnitelmassa luetelluista riskeistä toteutui, muttei niillä ollut vaikutusta projektin etenemiseen. Projektisuunnitelmaan oli hyvin listattu erilaiset riskit, eikä projektin aikana havaittu riskejä, jotka olisivat jääneet huomioimatta.

Määrittelydokumentti

Ensimmäisen iteraation määrittelydokumentti oli melko lyhyt, mutta kuvasi kuitenkin hyvin ohjelmistolta ensimmäisessä toteutusvaiheessa vaaditut ominaisuudet. Dokumentti määritteli tarkasti ensimmäisen iteraation tuloksena syntyvän ohjelmiston. Määrittelydokumentissa kiinnitettiin ohjelmiston toteutuskieleksi pelkästään MATLAB.

Suunnitteludokumentti

Suunnitteludokumentissa selostettiin yleiset periaatteet ohjelmointityylistä ja kuvattiin lyhyesti laskentamenentelmät. Lisäksi käyttöliittymän ulkonäöstä esitettiin alustava kuva ja hahmoteltiin erilaiset komponentit valmiissa ohjelmassa. Lisäksi määriteltiin syöte-, tulos-, ja tietokantatiedostojen muoto.

Testausdokumentti

Ensimmäisen iteraation testausdokumentti jäi mielestämme hieman vajaaksi, joten päätimme käyttää enemmän tarmoa seuraavan kierroksen testausdokumenttiin.

3.2 Toinen iteraatio

Määrittelydokumentti

Toisen iteraation määrittelydokumenttissa korjattiin määrittelyt vastaamaan demotilaisuudessa asiakkaalta saamaamme palauttetta. Aikaisemmissa dokumenteissa kuvattu mahdollisuus asettaa kynnysraja-arvo poistettiin Ari Rantasen ja Juho Rousun aloitteesta liian vaikeana toteuttaa. Lisäksi käyttöliittymä päätettiin toteuttaa englanninkielisenä, ja jätimme pois fragment-titietokannan käyttöliittymän, koska asiakkaan mielestä MATLAB:in sisäänrakennettu tekstieditori oli riittävä.

Suunnitteludokumentti
Toinen suunnitteludokumentti ei eronnut paljoakaan ensimmäisestä. Tässä dokumentissa vain korjattiin muutamia pieniä yksityiskohtia asiakkaan palautteen mukaan ja käyttöliittymän kuvina käytettiin oikeita kuvia ensimmäisen iteraation ohjelman käyttöliittymästä. Lisäksi moduulihierarkia kirjattiin vastaamaan paremmin ensimmäisellä kierroksella syntyneitä rakenteita.

Testaussuunnitelma
Testaussuunnitelmassa selvitimme eri moduuleille suunnitellut testit. Selvitimme myös käyttämiämme testaustapoja.

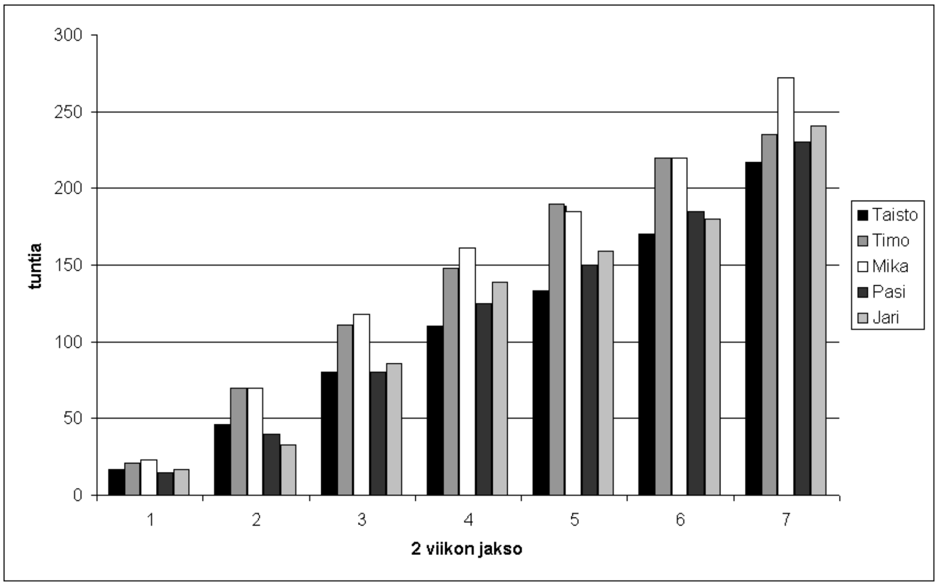
Testausdokumentti
Testausdokumentti sisältää testaussuunnitelman sisältämien testien tulokset.

Ylläpitodokumentti
Ylläpitodokumentti syntyi lyhyestä yhteenvedosta muista dokumenteista ja jokaisen ryhmän jäsenen omista funktioistaan tekemistä kuvauksista. Tässä dokumentissa listattiin myös globaalit muuttujat ja kerrottiin lyhyesti ohjelmiston toiminnoista.

Käyttöohje
Käyttöohjeen kirjoitti Timo katsoen hieman mallia englanninkielisestä PIDC-ohjelman käyttöohjeesta. Ohjelmamme on tavallaan uusi versio PIDC-ohjelmasta.

4 Työtunnit

Tehtyjä työtunteja tarkasteltiin kahden viikon jaksoissa. Näitä jaksoja oli 7 kpl (ja yksi puolikas). Ensimmäiseen iteraatioon käytettiin jaksot 1-4 ja toiseen 5-7. Ohjelmiston tekemiseen käytettiin yhteensä noin 1300 tuntia, joka tekee lähes 9 henkilötyökuukautta.



Kuva 1: Käytetyt työtunnit

5 Ohjelmisto

Ohjelmassa on 2100 riviä, joista 200 on kommenttirivejä ja 300 tyhjiä rivejä. Arvioimme projektisuunnitelmassa, että ensimmäisen iteraation jälkeen ohjelmassa on noin 1000 riviä MATLAB-koodia, ja toisen iteraation jälkeen 2000-2500 riviä myöhemmin päätettävillä työkaluilla kirjoitettua koodia. Arviossamme olimme varautuneet mahdollisesti toteuttamaan jotkin osat ohjelmistosta muilla työkaluilla kuin MATLAB:illa. Arvio on kuitenkin ihan hyvä, sillä ohjelmistossa on noin 1600 riviä todellista koodia, 2100 menee jopa arvioituun haarukkaan. Jos olisimme käyttäneet jotain muita työkaluja, olisi rivimäärä varmasti kasvanut.

Ohjelmisto sisältää 36 MATLAB:in m-tiedostoa ja yhden käyttöliittymään liittyvän fig-tiedoston. Näissä tiedostoissa on noin 50 funktiota. Lisäksi ohjelmistossa on muutamia testauksessa käytettyjä m-tiedostoja, joita itse ohjelma ei käytä. Käyttöohjeesta löytyy kuvia siitä, millaiselta ohjelmisto näyttää sekä Windows- että Linux-ympäristöissä.

Lähteet

[Ran02] Rantanen, A. et al., Computing Positional Isotopomer Distributions from Tandem Mass Spectrometric Data. Metabolic Engineering, hyväksytty.