Ohjelmistotuotantoprojekti “Bugfind”
Projektisuunnitelma

Project group
Perttu Haimi, Petteri Hintsanen, Philipp Masche, Toni Ruokolainen and Osku Salerma

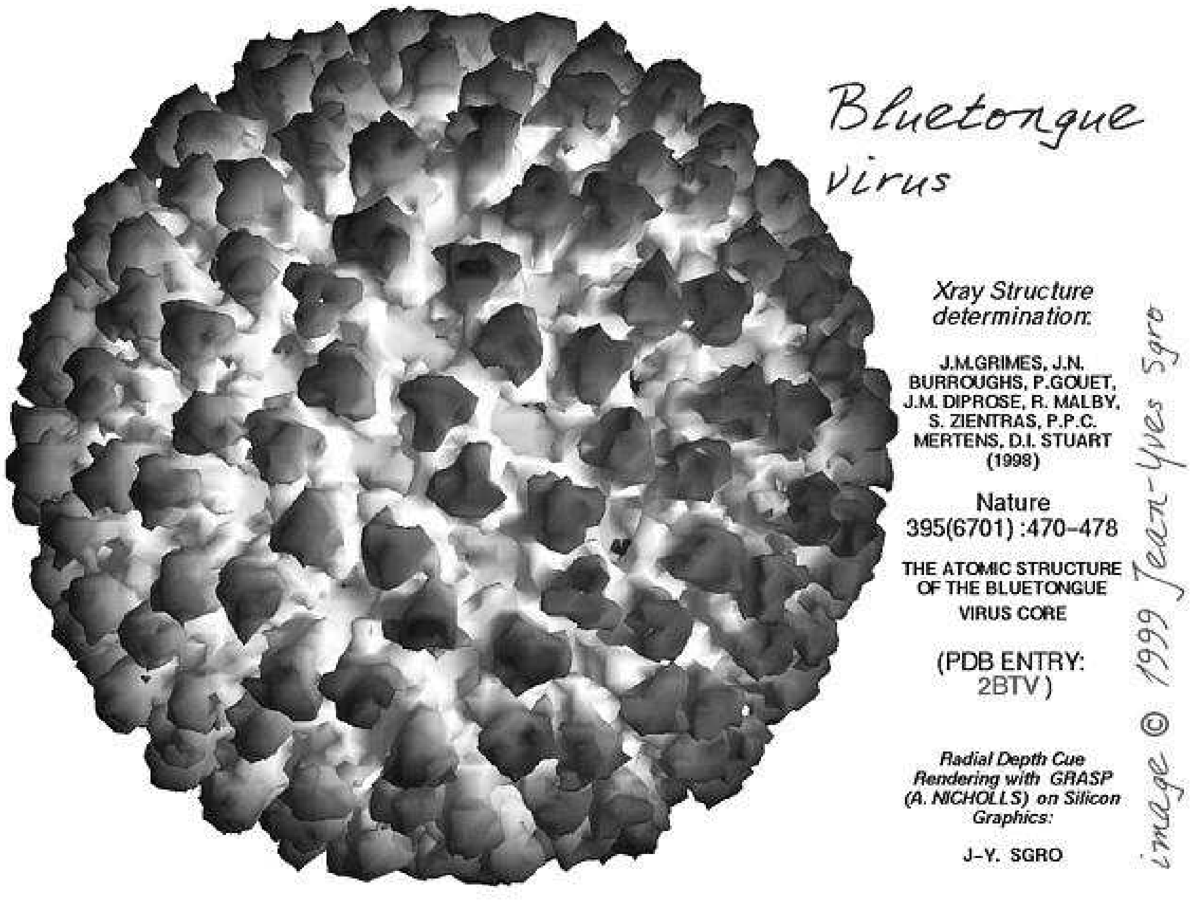
Instructor
Peter Lamberg

Helsinki 11. maaliskuuta 2001
Projektisuunnitelma
UNIVERSITY OF HELSINKI
Department of Computer Science

Versiohistoria

- v0.1, 24.1.2001 (vedos)
- v0.2, 29.1.2001 (vedos)
- v1.0, 31.1.2001 (valmis dokumentti)
- v1.1, 8.2.2001 (ulkoasumuutoksia)
- v1.2, 10.3.2001 (ulkoasumuutoksia)

Sisältö



Bluetongue virus
Xray Structure determination:
J.M.GRIMES, J.N. BURROUGHS, P.GOUET, J.M. DIPROSE, R. MALBY, S. ZIENTRAS, P.P.C. MERTENS, D.I. STUART (1998)
Nature 395(6701) :470-478
THE ATOMIC STRUCTURE OF THE BLUETONGUE VIRUS CORE
(PDB ENTRY: 2BTV)
Radial Depth Cue Rendering with GRASP (A. NICHOLLS) on Silicon Graphics:
J-Y. SGRO
image © 1999 Jean-Yves Sgro

1 Yleiskuvaus

Projektin “Bugfind” tehtävänä on määritellä, suunnitella ja toteuttaa yksi keskeisistä viruksen rekonstruktioprosessissa käytettävistä ohjelmakomponenteista. Itse rekonstruktioprosessin tarkoituksena on mallintaa tietystä viruksesta kolmiulotteinen malli käyttämällä hyväksi viruksen kaksiulotteisia projektioita. Projektiot kuvataan elektronimikroskoopilla. Rekonstruktioprosessi sisältää laskennallisesti hyvin vaativia osia, joten sen suorittamisessa hyödynnetään rinnakkaislaskentaa.

Rekonstruktioprosessi koostuu karkeasti ottaen kolmesta vaiheesta:

1. Virusten kuvaaminen elektronimikroskoopilla.
2. Virusten erottaminen otetusta kuvasta sekä niiden orientaation tunnistaminen (kuvat ovat siis kaksiulotteisia projektioita kolmiulotteisista viruksista).
3. Kolmiulotteisen virusmallin muodostaminen erotettujen projektioiden avulla.

Projektin tuottama ohjelmisto tulee sijoittumaan ylläolevassa listassa kohtaan 2. Nykyisessä järjestelmässä elektronimikroskooppikuvasta pystytään erottamaan symmetrisiä viruksia esittävät projektiot; tavoitteena on suunnitella ja toteuttaa ohjelmisto, joka kykenee erottamaan kuvasta myös epäsymmetrisiä viruksia. Etsittävästä (epäsymmetrisestä) viruksesta on syötteenä käytettävissä kolmiulotteinen malli, jota projisoidaan eri kulmista ja saatuja projektioita etsitään elektronimikroskooppikuvasta. Näiden projektioiden sijainnit elektronimikroskooppikuvassa ovat ohjelmiston tulosteet. Käyttäjä voi vaikuttaa hakuun parametreilla (esimerkiksi vaatimalla tavallista tarkempia hakuja).

2 Kokoarvio

Alla on esitetty ohjelman arvioitu koko korkean tason ohjelmointikielen koodiriveinä (*Lines of Code*, LOC). Arvio on karkea, koska aiempaa informaatiota (kokemuksia aiheesta, tilastoja) on niukasti.

Ohjelman osa	**LOC**
Pääohjelma	**800**
Syöteparametrien käsittely	100
Syötekuvan lukeminen	200
Annetun 3D-mallin lukeminen	200
Datan hajautuksen valmistelu	200
Muuta koodia	100
Hajauttaminen	**600**
Datan lähetys solmuille	200
Datan vastaanottaminen solmuilta	200
Muuta koodia	200
3D-projisointi	**1000**
Haku	**800**
Fourier-analyysi	200
Vertailut	200
Muuta koodia	200
Tulosten käsittely	200
Yhteismäärä	**3200**

3 Ositus

Projektin vaihejakoon käytetään vesiputousmallia. Vesiputousmalliin päädyttiin ensisijaisesti siitä syystä, että annettu tehtävä on melko hyvin määritelty ja rajattu. Lisäksi projektilla ei ole kiire siinä mielessä, että asiakkaalle pitäisi saada nopeasti jotain näkyvää aikaiseksi. Vesiputousmallin sisältämät riskit (esimerkiksi yllättävät ongelmat toteutusvaiheessa) pyritään välttämään huolellisella vaatimusanalyysillä ja suunnittelulla sekä kokeilemalla käytännössä hankalalta tuntuvia ratkaisuja jo suunnittelun aikana. Siten esimerkiksi kahden osavaiheen jako ei ole jyrkkä, vaan osavaiheet voivat mennä jonkin verran päällekkäin.

Projektimalliksi sopisi myös spiraalimalli, mutta tähän annettu aikataulu on ilmeisen riittämätön: spiraalimalli vaatii runsaasti tarkkaa analysointia ja dokumenttien tuottamista jokaisella iteraatiokierroksella. Annetuissa puitteissa iteraatioiden määrä jäisi melko pieneksi, joten spiraalimallin hyödyntämisestä ei olisi sanottavasti etua.

Vesiputousmallin mukaisesti projekti koostuu seuraavista osista:

1. **Vaatimusanalyysi**: Vaatimusanalyysissä kartoitetaan ohjelmistolta vaadittavat ominaisuudet, ohjelmiston käyttäjät ja sidosryhmät, liittymät sekä rajoitteet. Vaatimusanalyysin tuloksena saadaan *määrittelydokumentti*, jota käytetään suunnittelutyön pohjana.

2. **Suunnittelu**: Suunnitteluvaiheessa laaditaan yksityiskohtainen ohjelmiston toteutussuunnitelma vaatimusanalyysistä saadun määrittelydokumentin avulla. Tuloksena saadaan *suunnitteludokumentti*, joka sisältää ohjelmiston toteuttamisessa tarvittavat yksityiskohtaiset tiedot. Projektissa käytetään perinteisiä proseduraalisia suunnittelumenetelmiä. Suunnittelun aikana voidaan tehdä pienimuotoisia käytännön testejä mahdollisten epäselvyyksien selvittämiseksi. Suunnittelun aikana kirjoitetaan myös *testaussuunnitelma*, joka sisältää valmiille ohjelmistolle suoritettavat moduuli-, integrointi- ja validointitason testit.

3. **Toteutus**: Toteutusvaiheessa suunniteltu ohjelmisto toteutetaan suoraviivaisesti edellisessä vaiheessa saadun suunnitteludokumentin avulla. Ohjelmointikielenä on C. Toteutusvaiheessa tehdään samalla jo testausta sopivissa tilanteissa testaussuunnitelman mukaan, eli aloitetaan *testausdokumentin* kirjoittaminen. Toteutuksen aikana kirjoitetaan myös *toteutusdokumentti*, joka sisältää mahdolliset poikkeukset aiempiin suunnitelmiin ja määrityksiin.

4. **Testaus**: Varsinaisessa testausvaiheessa valmista ohjelmistoa testataan mahdollisimman kattavasti testaussuunnitelman mukaisesti ja samalla täydennetään toteutusvaiheessa aloitettua testausdokumenttia. Testausvaiheen päätteeksi saadaan valmis testausdokumentti.

5. **Projektin päättäminen**: Projektin lopussa kirjoitetaan vielä *loppuraportti*, johon kootaan yhteenveto projektista, työtunneista yms. projektin kannalta olennaisista asioista. Tuotettu ohjelmisto ja dokumentaatio viimeistellään vielä luovutuskuntoon, mikäli tarpeen. Mitään funktionaalisia muutoksia ohjelmistoon ei enää tehdä.

Näiden vaiheiden lisäksi projektin tuotoksena syntyy *käyttöohje*, *ylläpitodokumentti*, *toteutusympäristön kuvaus*, projektiryhmän *pöytäkirjat*, *muistiot* ja *työtuntilistat* sekä dokumentti *muutoksista jäädytettyihin dokumentteihin*. Näitä dokumentteja kirjoitetaan tasaisesti projektin keston aikana.

4 Aikataulu

Allaoleva aikataulu sisältää eri osavaiheille varatut ajat projektin kokonaisajasta. Oikeanpuolimmainen sarake (*jäädytys*) kertoo, milloin osavaihe on lopullisesti valmis, eli osavaiheen tuotoksena syntynyt dokumentti jäädytetään.

Osavaihe	**Viikkoja**	**Päivät**	**Jäädytys**
Projektisuunnitelma	2	19.1.–31.1.	(ei jäädytetä)
Vaatimusanalyysi	3	29.1.–18.2.	25.2.
Suunnittelu	5	19.2.–25.3.	1.4.
Toteutus	3	26.3.–15.4.	22.4.
Testaus	3	16.4.–6.5.	13.5.
Viimeistely	1	7.5.–13.5.	13.5.
Yhteensä	**17**	**19.1.–13.5.**	**13.5.**

5 Työnjako

Karkea työnjako on seuraava:

Hajauttaminen ja *MPI/MPIch*-yksityiskohdat	Osku ja Toni
Käytettävän *Beowulf*-klusterin ylläpito projektin aikana	Osku
Algoritmit (3D-projisointi, FFA, jne...)	Petteri ja Philipp
Projektin WWW-sivut	Perttu
Projektikokousten puheenjohtaja ja sihteeri	Philipp
Projektin aikataulussa pysymisen seuranta	Petteri

Kaikki projektin jäsenet osallistuvat dokumenttien kirjoittamiseen, suunnitteluun, toteuttamiseen ja testaukseen. Ylläoleva työnjako määrittelee lähinnä päävastuualueet.