

VERSTAS-PROJEKTI

Vaatimusmäärittely

Antti Hakala

Tomi Laamanen

Paavo Nieminen

Jukka Toivanen

7.1.2003

Jyväskylän yliopisto

Tietotekniikan laitos

Versiohistoria

Versio	Pvm	Tekijä	Kuvaus
0.1	14.10.	AH, PN	Lisätty: kansilehti + tietoa alkuperäisestä kuvauksesta ym.
0.2	23.10.	AH, PN	Lisätty: Numerolan miettimät vaatimukset ja luku 4. Järjestelty hieman. Yleiskatsaus (1.4) muutettu.
0.3	25.10.	PN	<p>Muutettu:</p> <ul style="list-style-type: none"> - jokainen vaatimus pyritty saamaan "testattavampaan muotoon" - mahdollista vastata kysymykseen "toteutuuko vaatimus vai ei?" lopputuotetta käyttämällä. - Johdantokappaleen rakennetta muokattu <p>Lisätty:</p> <ul style="list-style-type: none"> - sisällysluettelo. - termi "ulkoinen suurtietokanta" vastaamaan malli- ym. gallerioita - Projektissa voi olla vain yhteensopivia tietoja prioriteetilla 3, koska käytännön toteutus saattaisi vaatia laajempaa remonttia . <p>Poistettu:</p> <ul style="list-style-type: none"> - vaatimus "yhden managerin tekemisestä", sillä sovelluksen kannalta vastaava toiminnallisuus toteutuu muiden vaatimusten summana. - Korvattu viittaus "välilehtiin" viittauksella työpöydän visuaalisen asemoinnin muuttumiseen
0.4	28.10.	JT	Poistettu: kuvauksen käyttöliittymä-osasta suorat viitaukset alisovellusten sisällyttämisestä yhteen ikkunaan. Lause "sovelluksen editorin voi vaihtaa käyttäjän haluamaksi" turhana.
0.5	29.10.	TL	Lisätty: määritelmät ja lyhenteet sekä viitteet ja liitteet. Ulkoasua muokattu.
0.6	30.10.	JT	<p>Muutettu: Vaatimusten priorisointi siten, että 1 vaatimukset ovat "huolellisesti testattavia". Siirretty vaatimusmäärittelyn muuttaminen omaksi luvukseen.</p> <p>Lisätty: Vaatimukset hakutoiminnosta ja näkymien reaaliaikaisesta päivittämisestä.</p>
0.7	04.11.	TL	Muutettu: Ulkoasua muokattu.

Tekijöiden lyhenteet:

- AH Antti Hakala
- TL Tomi Laamanen
- PN Paavo Nieminen
- JT Jukka Toivanen

Hyväksytty

Pvm Allekirjoitus

Pvm Allekirjoitus

Sisältö

1	Johdanto	1
1.1	Tarkoitus	1
1.2	Sovelluksen päämäärä	1
1.3	Määritelmät ja lyhenteet	1
1.3.1	Sovellukset, kirjastot ja tiedostot	1
1.3.2	Ohjelmointikielet ja -ympäristöt	2
1.3.3	Mallinnusprosessi ja laskenta	2
1.3.4	Verstas	3
1.4	Yleiskatsaus	3
2	Kuvaus	4
2.1	Järjestelmän kuvaus	4
2.1.1	Alisovellusten hallinta	4
2.1.2	Tiedostojen hallinta	4
2.1.3	Käyttöliittymä	4
2.2	Käyttäjä	4
2.2.1	Perusmallintaja	5
2.2.2	Eksperttimallintaja	5
2.2.3	Mallien kehittäjä	5
2.3	Vaatimusten jakaminen	5
3	Sovelluksen vaatimukset	6
3.1	Ulkoiset rajapinnat	6
3.1.1	Käyttöjärjestelmärajapinnat	6
3.1.2	Tietoliikennerajapinnat	6
3.1.3	Tietokannat	6
3.2	Alisovellusrajapinnat	6
3.3	Toiminnalliset vaatimukset	6
3.4	Suunnittelurajoitukset	6
3.5	Muita vaatimuksia	7
3.5.1	Käyttöliittymä	7
3.5.2	Alisovellusten tiedot	7
3.5.3	Sovelluksen asennus	7
3.5.4	Numerrin-hankkeen jatkokehityksen huomioiminen	7
4	Projektin muut vaatimukset	8
4.1	Editori	8
4.2	Fortran-kääntäminen ja laskentasovelluksen suorittaminen	8
5	Vaatimusmäärittelyn muuttaminen	9

1 Johdanto

1.1 Tarkoitus

Tämä dokumentti määrittelee Verstas-projektissa tuotettavan sovelluksen ominaisuudet ja toiminnallisuudet sekä sovelluksen käyttöympäristölleen asettamat vaatimukset. Luku 1 esittelee lyhyesti dokumentin rakenteen, Verstas-projektin taustan ja dokumentissa käytettävät termit. Luku 2 perustelee myöhemmin esitettäviä vaatimuksia: Se kuvaa Verstas-sovellusta arvioituine käyttöympäristöineen ja tavoitteineen ja sisältää tilaajan alkuperäisiä vaatimusesityksiä. Luku 3 määrittelee sovellukselle täsmälliset vaatimukset, joiden toteutuminen voidaan yksikäsitteisesti testata lopputuotetta käyttämällä. Luvussa 4 määritellään vaatimuksineen Verstas-sovelluksen ulkopuoliset osat, jotka Verstas-projekti toteuttaa.

1.2 Sovelluksen päämäärä

Numerolassa kehitetyn Numerrin-simulointiympäristön eri ohjelmistot kootaan yhdeksi mallinnusympäristöksi eli saman työpöydän alla toimivaksi kokonaisuudeksi.

1.3 Määritelmät ja lyhenteet

1.3.1 Sovellukset, kirjastot ja tiedostot

Alisovellus: Numerolan toteuttama sovellusohjelma, jolla toteutetaan yksi mallinnusprosessin vaiheista.

Dmg: Alisovellus, joka muuttaa 2D-geometrian laskentaverkoksi.

Emacs: Laajennettava ja muokattava tekstieditori.

Exodus: Alisovellus, työkalu 3-ulotteisen laskentageometrian luomiseen.

Genesis: Alisovellus, työkalu 2-ulotteisen laskentageometrian luomiseen.

Makefile: Make-sovelluksen käyttämä tiedosto ohjelman lähdekoodin kääntämisen hallintaan.

Mbgen: Alisovellus, joka muuttaa 3D-geometrian laskentaverkoksi.

Numerrin-ydin: Numerolan toteuttama FEM-aliohjelmakirjasto.

Plot: Alisovellus visualisointiin.

TeeKL, AutoUI: Alisovellus, työkalu graafisen käyttöliittymän automaattiseen tuottamiseen ja hallintaan.

Verkkogeneraattori: Alisovellus, joka tuottaa geometriasta laskentaverkon. Verstaan alisovellusten käytössä on useita eri tilanteisiin sopivia verkkogeneraattoreita, kuten Verkotin, Dmg, Mbgen.

1.3.2 Ohjelmointikielet ja -ympäristöt

Elisp: Emacs lisp, Emacsin laajentamiseen käytettävä ohjelmointikieli.

Fortran: Erityisesti laskennallisten ongelmien ratkaisuun käytetty ohjelmointikieli.

Kylix: C++/Object Pascal -kehitysympäristö Linux-käyttöjärjestelmälle.

Object Pascal: Olio-ohjelmointilaajennos Pascal-ohjelmointikieleen.

1.3.3 Mallinnusprosessi ja laskenta

FEM: Finite Element Method, äärellisten elementtien menetelmä, eräs numeerinen menetelmä osittaisdifferentiaaliyhtälöiden numeeriseksi ratkaisemiseksi.

Geometrian luonti: Mallinnusprosessin vaihe, jossa tutkittavan fysikaalisen järjestelmän ominaisuudet pyritään kuvaamaan tietokoneohjelman avulla. Tuloksena on ns. laskentageometria.

Laskenta: Mallinnusprosessin vaihe, jossa käännetty laskentaohjelma ajetaan syötteenään laskentaverkko sekä parametrit, tulosteena saadaan laskentatulokset.

(Laskenta)geometria: Tietokoneella luotu tutkittavan fysikaalisen kappaleen muodon ja rakenteen kuvaus, esitettynä joukosta yksinkertaisia käyriä tai pintoja muodostuvana approksimaationa.

(Laskenta)geometrian luonti: Mallinnusprosessin vaihe, jossa kuvataan tutkittavan fysikaalisen järjestelmän geometriset ominaisuudet. Tuloksena on laskentageometria.

(Laskenta)malli: Matemaattinen malli, jonka ilmenemismuoto on fortran-lähdekoodi.

Laskentaohjelma: Fortran-kääntäjällä laskentamallista käännetty ajettavissa oleva ohjelma.

Laskentatulos: Laskentaohjelman tuloste, matemaattisen mallin ratkaisu annetuilla lähtötiedoilla (geometria, parametrit), käytännössä ASCII-tekstinä esitettyä numeerista dataa.

Laskentaverkko: Verkkogeneraattorin tuloste, geometrian jako laskentaa varten osiin esim. kolmioihin tai tetraedreihin.

Laskentaverkon luonti: Mallinnusprosessin vaihe, jossa laskentageometriasta tuotetaan verkkogeneraattorilla laskentaohjelman tarvitsemaan laskentaverkko.

Mallin toteutuksen etsintä: Valittua matemaattista mallia vastaavan ohjelman etsiminen tai luominen.

Mallin valinta: Mallinnusprosessin vaihe, jossa päätetään (yleensä kirjallisuuteen tai tutkimukseen nojaten) myöhemmissä vaiheissa käytettävä matemaattinen malli.

Mallinnusprosessi: Nykyaikaisen tuotekehityksen ja tutkimuksen osa-alue, jossa prototyypin rakentamisen ja kokeellisen mittaamisen sijasta tehdään kokeiluja matemaattisella mallilla, joka kuvaa järjestelmän toimintaa annetuissa olosuhteissa.

Optimointi: Simuloinnin (automaattinen) toistaminen jollakin kriteerillä geometriaa ja/tai parametreja muuttamalla, tavoitteena parempi lopputulos.

Parametrit: Laskentaohjelman syöte, esim. tutkittavan ilmiön fysikaaliset ominaisuudet.

Visualisointi: Laskentatulosten esitys ihmisen ymmärtämässä muodossa (käytännössä graafisesti erilaisin tavoin), eräs mallinnusprosessin vaihe.

1.3.4 Verstas

Mallinnusympäristö: Kokonaisuus, jossa kaikki mallinnuksen vaiheet voidaan tehdä yhtenäisen käyttöliittymän avulla hallittavilla työkaluilla.

Numerrin-Verstas: Verstas, Verstas-ympäristö. Mallinnusympäristö, jossa kaikki mallinnusprosessin vaiheet voidaan toteuttaa käyttäen Numerolan työkaluja ja Numerrin-ydintä.

Projekti: Kokonaisuus, joka sisältää kaikki tiettyyn simulointitehtävään liittyvät tiedot (s.o. mallit, geometriat, parametrit, tulokset, raportit jne).

Raportti: Tutkimus-, päivitys- tai muu raportti, esim. pdf-muodossa.

Tietokanta: Tässä projektissa hakemistorakenne sisältäen uudelleen hyödynnettäviä mallinnukseen käytettäviä tiedostoja.

1.4 Yleiskatsaus

Tämän dokumentin rakenne mukalee IEEE:n vaatimusmäärittelysuositusta IEEE Std 830-1998 [2].

2 Kuvaus

2.1 Järjestelmän kuvaus

Nykyinen Numerrin-simulointiohjelmisto koostuu useista alisovelluksista. Verstaasovellus tulee kommunikoida erillisten alisovellusten kanssa ja huolehtimaan niiden yhteiskäytöstä sekä syöttö- ja tulostustiedostojen hallinnasta. Myöhemmin tullaan tekemään uusia alisovelluksia liitettäväksi Verstaaseen, joten kommunikoinnin suunnittelussa on huomioitava myöhempi laajennettavuus.

2.1.1 Alisovellusten hallinta

Alisovellusten täytyy olla Verstaan avulla käyttäjän helposti kontrolloitavissa siten, että alisovellusten tarjoamia ominaisuuksia voi käyttää yhteistyössä Verstaan kanssa.

2.1.2 Tiedostojen hallinta

Alisovellukset ottavat vastaan syötetiedostoja ja tulostavat tiedostoja, jotka jälleen menevät syötteeksi uusille alisovelluksille. Tärkeä osa Verstaan suunniteltua toiminnallisuutta on tiedostojen hallitseminen ja liittäminen asiaankuuluviin alisovelluksiin.

2.1.3 Käyttöliittymä

Mallin ympäristöä kaavaillaan kaupalliseksi tuotteeksi, jonka olennaisia ominaisuuksia ovat käytettävyys ja visuaalinen ilme. Verstaan ulkoasu tulee olla yhtenäinen ja helposti hallittava.

Työpöytä jaetaan eri (esim. välilehdin toteutettaviin) näkymiin. Jako voisi esimerkiksi olla:

1. Simulointimallin esikäsittely: Näkymässä geometrian luontiin tarvittavat työkalut (olemassa valmiit sovellukset 2D-verkon sekä 3D-verkon tekoon)
2. Matemaattisen mallin määrittely: Näkymässä esimerkiksi editori (joko emacs tai muu valinnainen editori), komentorivi tai/ja painikkeet tarvittaville käyttöjärjestelmäkomennolle (sisällyttävä ainakin ohjelman kääntäminen) ja esiohjelmoidut simulointimallit sisältävä tietokanta.
3. Simulointimallin käyttö: Näkymässä suoritusnappi ja/tai komentorivi, suorituksen aikainen tekstitieto ja mahdollinen graafinen tulostus
4. Simulointitulosten jälkikäsittely: Näkymässä visualisointityökalut (olemassa oleva Plot-ohjelma) sekä komentorivi ja/tai toimintapainikkeita.

2.2 Käyttäjä

Verstaan suunniteltu käyttäjä tulee olemaan *perusmallintaja*, *eksperttimallintaja* tai *mallien kehittäjä*. Käyttäjä ei välttämättä tunne käyttöjärjestelmän toimintaa syvällisemmin.

2.2.1 Perusmallintaja

Perusmallintaja käyttää suhteellisen rajattua määrää malleja ja geometrioita. Hän muuttelee hieman geometriaa ja parametreja, mutta ei juuri koske itse laskentamalliin. Perusmallintaja myös tekee paljon raportteja asiakkailleen.

2.2.2 Eksperttimallintaja

Eksperttimallintaja saattaa tehdä tutkimusta tai hankalampia tilausmallinnuksia. Hän luo monimutkaisempia geometrioita ja saattaa muutella myös laskentaohjelman koodia. Eksperttimallintajan raportit ovat yksityiskohtaisempia ja teoreettisempia kuin perusmallintajan.

2.2.3 Mallien kehittäjä

Mallien kehittäjä on kiinnostunut uusien laskentamallien ohjelmoinnista ja olemassa olevien parantelusta. Hän käyttää geometrioita, parametreja ja visualisointia lähinnä testatakseen mallinsa toimivuutta. Mallien kehittäjän raportointi keskittyy ohjelmien ja algoritmien kommentointiin.

2.3 Vaatimusten jakaminen

Jokaiselle vaatimukselle on annettu prioriteettinumero 1-3, jossa 1 on prioriteetiltään korkein. Numero on annettu suluissa heti vaatimuksen esittelyn jälkeen. Projektin tuottama *sovellus katsotaan toteutetuksi, kun prioriteetilla 1 ja 2 merkityt vaatimukset toteutuvat*. Prioriteetin 1 vaatimusten tulee olla huolellisesti testattuja ja toimivia. 3:lla merkityt vaatimukset toteutetaan aikaresurssien puitteissa ja niistä tullaan tarpeen mukaan luopumaan 1:llä ja 2:lla merkittyjen eduksi.

3 Sovelluksen vaatimukset

3.1 Ulkoiset rajapinnat

3.1.1 Käyttöjärjestelmärajapinnat

- sovellus on toimii Linuxissa X-Window käyttöympäristössä (1)
- sovellus on toimii Windows-alustalla (3)

3.1.2 Tietoliikenne-rajapinnat

- Sovellus käyttää TCP/IP-protokollaa tietoliikenneprotokollana kommunikoidessaan alisovellusten kanssa. (1)
- Sovellus käyttää salattua tietoliikennettä (3)

3.1.3 Tietokannat

- Sovelluksen käyttäjä voi tallentaa lokaalisti työtietokantaan mallinnuskomponentteja (1)
- Käyttäjä voi tuoda mallinnuskomponentteja ulkoisesta suurtietokannasta eli ns. gallerioista (1)
- Käyttäjä voi tehdä gallerioihin avainsanoihin perustuvia hakuja (1)
- Samaa ulkoista suurtietokantaa voi käyttää useampi Verstas-sovellus samanaikaisesti (1)
- Sovellus osaa yhdistää tietyn tyyppisen tiedoston oikeaan alisovellukseen tiedostopäätteen perusteella (1)

3.2 Alisovellusrajapinnat

Verstaan ja alisovellusten välisessä kommunikaatiossa käytettävä protokolla määritellään erikseen.

3.3 Toiminnalliset vaatimukset

Kts. liite Käyttötapaukset. Dokumentissa esitetyt käyttötapaukset toteutuvat sovelluksen toiminnassa.

3.4 Suunnittelurajoitukset

- Kylix 3 Professionalin ominaisuudet riittävät sovelluksen toteuttamiseksi

3.5 Muita vaatimuksia

3.5.1 Käyttöliittymä

Käyttöliittymän ulkoasu ja toiminnallisuus määritellään prototyypin avulla kolmessa iteraatiossa siten, että vaatimukset kunkin iteraation prototyypeille määritetään erikseen (kts. Verstas-projektin kokospöytäkirjat). Käyttöliittymän täytyy lisäksi toteuttaa seuraavat vaatimukset:

- Käyttäjä voi mukauttaa työympäristöä, ja asetukset tallentuvat projektikohtaisesti (1)
- Verstas tarjoaa käyttäjäryhmien mukaan esimääriteltäviä mukautusasetuksia (3)
- Sovelluksessa on avustus-toiminto ja käyttöopas (1)
- Alisovellusten ikkunat pysyvät yhden pääikkunan sisällä (3)

3.5.2 Alisovellusten tiedot

- Käyttäjä voi sovelluksen kautta käskä alisovellusten tietoja tallentumaan ja muutoksia päivittämään (1)
- Käyttäjä voi käskä alisovelluksia päivittämään näkymänsä muuttuneiden tietojen mukaisiksi (1)
- Käyttäjä voi määrätä, että mikäli alisovellus käyttää jotakin tiedostoa, se päivittää näkymänsä aina tiedoston muuttuessa (2)
- Projektissa voi olla vain yhteensopivia tietoja (3)

3.5.3 Sovelluksen asennus

- Sovelluksessa on mukana asennuspaketti (2)
- Sovelluksessa on mukana asennuspaketti, joka sisältää myös alisovellusten asennuksen (3)

3.5.4 Numerrin-hankkeen jatkokehityksen huomioiminen

- Sovelluksen lähdekoodia ei tarvitse muuttaa alisovelluksen muuttuessa, ellei rajapinta vaadi laajentamista (1)
- Sovelluksen lähdekoodia ei tarvitse muuttaa uuden alisovelluksen lisäämiseksi, ellei rajapinta vaadi laajentamista (2)
- Sovelluksen lähdekoodia ei tarvitse muuttaa erilaisten datatiedostojen määrän tai tiedostotarkentimien merkityksen muuttuessa (2)

4 Projektin muut vaatimukset

Verstas-sovelluksen lisäksi projektin on toteutettava määriteltävän rajapinnan toteuttavat alisovellukset, jotka täyttävät seuraavat vaatimukset:

4.1 Editori

- Verstaan käytössä on tekstieditori (1)
- Projekti toteuttaa Lisp-ohjelman, jonka avulla Emacs-editori toteuttaa alisovellusrajapinnan (1)
- Editorin voi vaihtaa miksi tahansa alisovellusrajapinnan toteuttavaksi editoriksi (2)

4.2 Fortran-kääntäminen ja laskentasovelluksen suorittaminen

Verstas-projekti tuottaa alisovelluksen, joka toteuttaa seuraavat vaatimukset:

- Alisovellus toteuttaa alisovellusrajapinnan ja on siten käynnistettävissä Versta-sovelluksesta (1)
- Alisovellus osaa tehdä annettujen tiedostojen ja muiden tietojen perusteella syntaktisesti oikeanlaisen Numerrin-pakettia käyttävän makefile-tiedoston (1)
- Käyttäjä voi editoida alisovelluksen tekemää makefileä (1)
- Alisovellus osaa käynnistää make-ohjelman parametrinaan tehty makefilen (1)
- Alisovellus ilmoittaa mahdollisista käännösvirheistä (1)
- Alisovellus osaa käynnistää käännetyn laskentaohjelman (1)
- Alisovellus näyttää käyttäjälle käytön aikana tehdyt tulosteet (standard output / standard error) (1)
- Alisovellus osaa keskeyttää käynnistetyn laskentaohjelman hallitusti (1)
- Alisovellus osaa keskeyttää käynnistetyn laskentaohjelman pakotetusti (1)
- Alisovellus osaa ilmoittaa laskennan epänormaalia keskeytymisestä (3)

5 Vaatimusmäärittelyn muuttaminen

Tätä vaatimusmäärittelyä voidaan muuttaa projektin kuluessa seuraavasti:

- Jos projektin aikana tulee esille muita vaatimuksia, jotka ovat oleellisia sovelluksen toiminnallisuuteen ja ovat toteutettavissa projektin puitteissa, niistä sovitaan erikseen
- Jos projektin aikataulu ei riitä kaikkien (1) vaatimusten toteuttamiseen, niistä on keskusteltava ja sovittava erikseen

Viitteet

- [1] Santanen Jukka-Pekka, *Opinnäytteiden kirjoittaminen, lyhyt oppimäärä*, saatavilla WWW-muodossa
<URL: <http://www.mit.jyu.fi/santanen/info/kirjoittamisesta.html>>, viitattu 29.10.2002.
- [2] Software Engineering Standards Committee of the IEEE Computer Society, *IEEE Recommended Practice for Software Requirements Specifications IEEE Std 830-1998*, 1998.

Liitteet

Liite 1. Käyttötapaukset