

## **Tahiti — tähtitieteellisten havaintojen tietokanta**

Tomi Hänninen  
Juho Muhonen  
Ismo Puustinen  
Kai Pääsky  
Pekka Simola  
Nuutti Varis

Helsinki 12.5.2003

Toteutusdokumentti

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

Tiedekunta/Osasto — Fakultet/Sektion — Faculty		Laitos — Institution — Department	
Matemaattis-luonnontieteellinen		Tietojenkäsittelytieteen laitos	
Tekijä — Författare — Author			
Tomi Hänninen Juho Muhonen Ismo Puustinen Kai Pääsky Pekka Simola Nuutti Varis			
Työn nimi — Arbetets titel — Title			
Tahiti — tähtitieteellisten havaintojen tietokanta			
Oppiaine — Läroämne — Subject			
Tietojenkäsittelytiede			
Työn laji — Arbetets art — Level		Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Toteutusdokumentti		12.5.2003	4 sivua
Tiivistelmä — Referat — Abstract			
Versiohistoria:			
1. Versio 1.0 (Aito ja alkuperäinen toteutusdokkari)			
Avainsanat — Nyckelord — Keywords			
Tahiti, fotometria, Standard Asteroid Photometric Catalogue			
Säilytyspaikka — Förvaringsställe — Where deposited			
Muita tietoja — övriga uppgifter — Additional information			
Versio 1.0			

## Sisältö

<b>1 Johdanto</b>	<b>1</b>
<b>2 Tietokanta</b>	<b>1</b>
<b>3 Tahiti–Api</b>	<b>1</b>
<b>4 Käyttöliittymä</b>	<b>3</b>

## 1 Johdanto

Tässä dokumentissa kuvataan suunnitteludokumentissa esitettyyn suunnitelmaan tulleet muutokset toteutusvaiheen aikana. Dokumentin tarkoituksena on kuvata tapahtuneet muutokset samalla tasolla, kuin järjestelmä on suunnitteludokumentissa kuvattuna. Tarkempi kuva järjestelmän sisäisistä toteutusratkaisusta selviää ohjelmiston lähdekoodista.

Järjestelmän yleisrakenne muuttui vain siten, että Atlas-siirtimen lisäksi toteutettiin myös erillinen Trajectory-siirrin (pakkaus `fi.helsinki.cs.group.tahiti.trajectoryimporter`), jolla järjestelmään saadaan siirretty erittäin suuria määriä ratatietoja yhdellä kerralla. Siirrin on yksikertainen jäsentäjä, joka siirtää parametrin saamansa ratatietoinformaation Tahiti-Apille, jonka kautta varsinainen ratatietolisääminen tapahtuu.

## 2 Tietokanta

Tietokanta pysyi lähes muuttumattomana suunnitteludokumentin versio 1.3 jälkeen. Alla on listattu kantaan vielä dokumentin jälkeen tulleet muutokset:

- Lightcurve-taulu sai `defaultTrajectory`n lisäksi `defaultEarthTrajectory` attribuutin, joka kertoo valokäyrän maan oletusrataelementit.
- LightcurveUpdate-näkymä laajennettiin näyttämään uusien oletusrataelementtien arvot.
- Funktio `default_trajectory` toimii hieman eri tavoin: Oletusrataelementti asteroidille  $a$  ajankohdalla  $t$  saadaan, kun valitaan se asteroidin  $a$  rataelementti, jonka epoch  $e$  on suurin niistä rataelementeistä, joille  $e \leq t$ . Jos tällaista ei ole, valitaan aikaisin rataelementti. Jos edellämainituin keinoin saadaan useampia rataelementtejä kuin yksi, valitaan näistä viimeisimpänä syötetty.
- Rataelementtien ja havaintopisteiden ajat ovat nyt aikoja ilman aikavyöhykettä. Ne oletetaan GMT-vyöhykkeen ajoiksi.
- Kanna vakioaikavyöhyke on nyt GMT.

## 3 Tahiti-API

Tahiti-APIin ei kohdistunut suunnitteludokumentin valmistumisen jälkeen paljoakaan muutoksia. Muutosten vähäisen määrän vuoksi ne kaikki ovat kuvattuina yhdessä luvussa. Tahiti-APIin ja sen alla oleviin käsittelijäluokkiin kohdistuneet muutokset ovat seuraavia:

- Lokikirjoituksesta mainitaan suunnitteludokumentissa, että `Entry`-kentän sisällä kentät erotetaan toisistaan `||`-merkkijonolla, ja näiden kenttien sisällä käytetään erottimena pilkkua. Pilkku ei kuitenkaan yleisyytensä vuoksi sovellu erottimeksi, ja oikeana erottimena käytettiin dollarimerkkiä.

- Tahiti–Apiin lisättiin toiminto, joka kestää enemmän kuin yhden rekistörintipyynnön tekemisen istunnon aikana, ellei käyttäjä ole tasoltaan ylläpitäjä. Toinen tällainen uusi ominaisuus, joka Tahiti–Apiin lisättiin oli kirjautumisyritysten rajoittaminen kolmeen yritykseen istunnossa. Mikäli kirjautuminen onnistuu, nollataan tämä sisäinen laskuri.
- Järjestelmän sisäiset järjestysnumerot muuttuivat int–tietotyyppistä long–tyyppiin. Tämä muutos vaikutti kaikkiin Tahiti–Apin ja käsittelijäluokkien metodeihin, joissa parametreinä käytettiin jotain sisäistä järjestysnumeroa. Näitä järjestysnumeroita ovat siis seuraavat:
  1. AsteroidID, asteroidin järjestelmän sisäinen järjestysnumero
  2. lightcurveID, valokäyrän järjestelmän sisäinen järjestysnumero
  3. userID, käyttäjän järjestelmän sisäinen järjestysnumero
- Valokäyrien lisäysmetodeihin tuli uusi Atlas–syöttimeen liittyvä parametri, atlasAspect. Metodien ottamiin parametreihin se vaikuttaa siten, että LightcurveHandler–luokan insertLightcurve–metodin viimeiseksi parametriksi tuli double

atlasAspect. Normaaliin insertLightcurve Api–kutsuun tämä vaikuttaa siten, että käsittelijäluokan metodiin kohdistuvaan kutsuun tuli viimeiseksi parametriksi uusi null–viite. Vastaava parametri lisättiin myös insertLightcurveAtlas Api–kutsuun, missä Api vain välittää uuden parametrin varsinaisen työn tekeväälle käsittelijäluokalle.

- DBControl–luokka hakee tietokannan yhteystiedot ja käyttäjien tunnukset ja salasanat tiedostosta TahitiDB.conf, joka on sijoitettu Tahiti–hakemiston juureen. Tiedoston muoto on sellainen, että jokaiselle riville sijoitetaan yksi tieto. Tietojen järjestys esitetään esimerkkien kera seuraavaksi (tiedostoon tulee siis vain vaaditut tiedot esitetystä järjestyksessä, ei alla mainittua järjestysluetteloa).

**Tietokannan osoite ja portit** jdbc:postgresql://alkokrunni.cs.helsinki.fi:10667/tahiti

**selaajan käyttäjätunnus** tahitibrowser

**selaajan salasana** brpassword

**syöttäjän käyttäjätunnus** tahitisubmitter

**syöttäjän salasana** ssalis

**ylläpitäjän käyttäjätunnus** tahitiadmin

**ylläpitäjän salasana** admininhankalasalasana

- TrajectoryHandler–luokan kuvauksessa mainitaan, että jokaisesta uudesta rataelementistä ja asteroidista tehdään merkintä järjestelmän lokiin. Näin ei kuitenkaan ole, vaan lisäyksistä kirjoitetaan lokiin vain yhteenvetotieto, jossa kerrotaan uusien rataelementtien ja asteroidien lukumäärä.
- UserHandler–luokkaa muutettiin siten, että käyttäjänumeroa 0, 'Internal' käsitellään erikoistapauksena. Tämä tarkoittaa sitä, ettei kyseistä käyttäjää esimerkiksi näy laisinkaan käyttäjähauissa, käyttäjän salasanaa ei pystytä vaihtamaan, eikä käyttäjällä voida kirjautua sisään.

- getUsers-metodi muuttui suunnitelmasta siten, että getUsers-metodille TahitiApi antaa viimeisenä parametrinä käsittelijäluokalla metodia käyttävän käyttäjän User-objektin. Tämän avulla käsittelijäluokka voi poistaa muiden käyttäjien kirjautumistunnukset palautettavasta informaatiosta, mutta jättää käyttäjän oman kirjautumistunnuksen näkyviin. Lisäksi getUsers metodeihin lisättiin ominaisuus, ettei metodi normaalisti palauta poistettuja, tason -1 käyttäjiä. Lisäksi hakukutsussa voidaan määritellä, että vain tietyn tason käyttäjät palautetaan. Tämä käyttäjätasoinformaatio löytyy UserQueryContainerista.
- AtlasImporter-luokka toimii seuraavilla järjestyksessä mainituilla parametreilla: käyttäjätunnus, salasana, atlas-tiedosto tai hakemisto, jossa tiedostot ovat; ja hakemisto, johon virheelliset (tai osittain virheelliset) atlas-tiedostot laitetaan. Jos parametrina on atlas-tiedosto-hakemisto, ohjelma käy hakemiston ja kaikki sen alihakemistot läpi. Jos taas parametrina on yksittäinen atlas-tiedosto, ohjelma käsittelee vain kyseisen tiedoston. Käyttöesimerkiksi: java tahiti.atlasimporter.AtlasImporter Käyttäjä munsalasana Atlastietohakemisto Virheellisetännehakemisto.

## 4 Käyttöliittymä

Lopulliseen versioon toteutettiin seuraavat luokat:

```

AsteroidDataBean
EventHistoryDataController
LoginBean
TahitiGraphPostProcessor
AsteroidDataController
Initializer
LoginController
UserInfoDataBean
AsteroidQueryBean
LightcurveDataBean
LoginDataBean
UserInfoUpdateController
AsteroidSubmitDataBean
LightcurveQueryBean
MailController
UserRegisterController
AsteroidSubmitDataController
LightcurveResultController
MailSubmitBean
UserRemoveController
CustomElementController
LightcurveSubmitBean
SiteConfigurationSubmitController
UserSearchBean

```

CustomElementSubmitBean  
LightcurveSubmitController  
SiteConfigurationSubmitDataBean  
UserSearchController  
EventHistoryDataBean  
LogQueryBean  
TahitiGraphBean  
UserSearchDataBean

Käyttäjakohtaista vain ylläpitäjille näkyvää kommenttikenttää ei toteutettu mahdollisten henkilö-tietolain aiheuttamien kysymysten takia. Toteutuksessa olisi myöskin ollut pieniä ongelmia. :-)

Sivulla asteroidsubmit.jsp oleva ratatietotiedoston lähettäminen ei toimi luotettavasti suurilla tiedostoilla. 1-2 Mt on käytännön maksimi. Toiminta saattaa parantua, jos Tomcat-palvelimen kekoa (heap) kasvatetaan joitakin megatavuja suuremmaksi kuin lähetettävän tiedoston koko on. Tämä tietysti varaa järjestelmän resursseja.