

## **Ylläpitodokumentti**

Boa Open Access

Helsinki 2.5.2006

Ohjelmistotuotantoprojekti

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

**Kurssi**

581260 Ohjelmistotuotantoprojekti (6 ov)

**Projektiryhmä**

Ilmari Heikkinen

Timo Hintsu

Erno Härkönen

Arto Vuori

Mikko Kautto

**Asiakas**

Olli Niinivaara

**Johtoryhmä**

Juha Taina

Riikka Kaven

**Kotisivu**

<http://www.cs.helsinki.fi/group/boa>

**Versiohistoria**

Versio	Päiväys	Tehdyt muutokset
1.0	2.5.2006	Ensimmäinen versio
0.1	26.4.2006	Alustava versio

# Sisältö

<b>1 Johdanto</b>	<b>1</b>
<b>2 Sanasto</b>	<b>1</b>
<b>3 Asennus ja käyttö</b>	<b>1</b>
<b>4 Koodin ylläpito</b>	<b>1</b>
4.1 Qriterium Statement-kielen laajentaminen . . . . .	1
4.1.1 Uusien elementtien lisääminen lauseisiin . . . . .	1
4.1.2 Uuden statement-olion lisääminen . . . . .	1
4.1.3 Uusien elementtien lisääminen headeriin . . . . .	1
4.2 Transformaattoreiden lisääminen . . . . .	2
4.3 DataLeecher-luokkien lisääminen . . . . .	2
4.4 Tulostimien lisääminen . . . . .	2
<b>5 Ehdotuksia jatkokehitystä varten</b>	<b>2</b>
5.1 Testaaminen . . . . .	3

# 1 Johdanto

Tämä on BOA Open Access-ohjelmistotuotantoprojektin ylläpidodokumentti. Ylläpidodokumentti on tarkoitettu ohjelmiston jatkokehittäjille ja ylläpitäjille. Dokumentissa käsitellään ohjelmakoodin ylläpitoa ja laajentamista, ja se sisältää myös ryhmän ehdotuksia jatkokehitystä varten. Dokumenttia ei ole tarkoitettu luettavaksi yksinään, vaan yhdessä suunnitteludokumentin ja ohjelmakoodin kanssa.

## 2 Sanasto

Tässä dokumentissa käytettävä sanasto on luettavissa BOA:n suunnitteludokumentista.

## 3 Asennus ja käyttö

BOA:n asennus- ja käyttöohjeet ovat luettavissa BOA:n käyttöohjeesta.

## 4 Koodin ylläpito

### 4.1 Qriterium Statement-kielen laajentaminen

#### 4.1.1 Uusien elementtien lisääminen lauseisiin

Lisättäessä uusia elementtejä QS-olioihin, elementti tulee lisätä ensin itse laajennettavaan Statement-luokkaan. Lisäksi muutoksia tarvitaan Statement-olion `getXMLDataFields()`-metodiin. Lisätyt elementit tulee lisätä myös `XMLProperties`-luokkaan, sekä Qriterium Statementin DTD:hen.

#### 4.1.2 Uuden statement-olion lisääminen

Uudet statement-oliot tulee toteuttaa Statement-luokan aliluokkina (yhteys-oliot toteutetaan tarkemmin `QConnection`-luokan alaluokkina, ja actor-tyyppiset oliot `QActor`-luokan aliluokkina). Lisäksi uusien statement-olioiden sisältämät elementit tulee lisätä `XMLProperties`-luokkaan.

#### 4.1.3 Uusien elementtien lisääminen headeriin

Uusien elementtien lisääminen headeriin tehdään samankaltaisesti kuin lisättäessä uusia elementtejä statement-olioihin. `QHeader`-luokassa on `getXMLDataElements()`-metodi, johon uudet elementit tulee lisätä. Lisäksi muutoksia pitää tehdä myös `XMLProperties`-luokkaan.

## 4.2 Transformaattoreiden lisääminen

Transformaattorien tulee toteuttaa Transformer-rajapinta. Rajapinta määrittelee void-metodin `transform(DataLeecher dataLeecher, QStatements qStatements)`. Transformaattori saa DataLeecher-oliolta `InputStream`-tyyppisen datavirran, josta syötettä voidaan lukea. Transformaattorin tulee lisätä syötteestä luetut oliot parametrina annettuun `QStatements`-olioon käyttämällä sen metodia `addQStatement(QStatement statement)`.

Uusi transformaattori tulee lisätä Boa:n pääohjelmaan, sekä sille pitää tehdä oma source-konfiguraatio (kts. käyttöohje).

## 4.3 DataLeecher-luokkien lisääminen

Uusien DataLeecherien tulee toteuttaa DataLeecher-rajapinta. DataLeecherin tulee tarjota transformaattoreille syötettä Javan `InputStream`-tyyppisenä oliona. Uusien DataLeecherien lisääminen vaatii myös muutoksia pääohjelmaan, tämän projektin puitteissa toteutettu pääohjelma tukee vain yhtä, levyltä tiedostoja lukevaa DataLeecheriä.

## 4.4 Tulostimien lisääminen

Boa-projektissa toteutettiin `Qriterium Statement` -tulostaja. Tulevaisuudessa voi tulla tarve myös muunlaiselle ulostuloformaatille. Tätä varten on olemassa `StatementPrinter`-rajapinta, jonka voi toteuttaa tilanteen vaatimalla tavalla. Esimerkiksi voisi olla tarve tulostaa `html`-muotoista dataa. Tämän saavuttamiseksi täytyisi luoda uusi luokka, joka toteuttaisi `StatementPrinter`-rajapinnan. Boa tukee useita samanaikaisia tulostajia. Halutut tulostajat täytyy rekisteröidä `QStatements`-luokan kuuntelijaksi `addStatementPrinter(StatementPrinter)`-metodilla. Jos tulostin tarvitsee uusia määreitä, ne tulee listätä Boa-pääohjelmaan.

## 5 Ehdotuksia jatkokehitystä varten

Osa ohjelmistolle vaatimusmäärittelyssä asetetuista ominaisuuksista jäi toteuttamatta aikataulun tiukkuuden takia. Hyviä jatkokehityksen kohteita ovat siis kaikki vaatimusdokumentissa mainitut toteuttamatta jääneet ominaisuudet.

Ohjelman toiminnallisuutta olisi hyvä laajentaa DataLeecher-rajapinnan toteuttavalla luokalla, joka hakisi transformaattoreille syötettä esim. `OAI-PMH`-protokollaa käyttäen suoraan metadatalähteestä, eikä syötedataa tarvitsisi täten hakea kiintolevylle.

`CiteSeerTransformer` on hyvin "nirso" syötteensä suhteen, ja monet `recordit` hylätään. Viallisista `recordeista` voisi yrittää saada luettua kaiken tiedon mitä niistä on luettavissa parantamalla transformerin vikasietoisuutta.

## 5.1 Testaaminen

Transformaattorien testit eivät testaa tuotettavia olioita tarpeeksi tarkasti, eikä niissä käytetty syötedata ole ollut kovin laajaa. Linkkien syntyminen oikein content-olioden välille on testattu järjestelmätestauksella tulostetta tutkien.