

# Xetor

Tietotekniikan sovellusprojekti

Jani Lirkki  
Olavi Paananen  
Raimo Pitkänen  
Jussi Räisänen

Sovellusraportti  
27. toukokuuta 2004  
Versio 1.0

Jyväskylän yliopisto  
Tietotekniikan laitos

# Xetor-projektin tietoja

- Tekijät:** Jani Lirkki (jaallirk@cc.jyu.fi)  
Olavi Paananen (laolpaan@cc.jyu.fi)  
Raimo Pitkänen (rapitkan@cc.jyu.fi)  
Jussi Räisänen (jtraisan@cc.jyu.fi)
- Yhteystiedot:** Työtila AgC224.1, puhelinnumero (014) 260 4967, sähköpostilistan osoite xetor@korppi.jyu.fi ja www-sivujen osoite <http://kotka.it.jyu.fi/xetor>.
- Työ:** Sovellusraportti tietotekniikan sovellusprojektiin.
- Tiivistelmä:** Xetor-projekti toteutti Jyväskylän yliopiston Informaatioteknologian tiedekunnalle XML-editorin. Tässä dokumentissa kuvataan määriteltyjen vaatimusten toteutumista sekä sovelluksen toteutusta ja toimintaa.
- Avainsanat:** XML, editori, Java, puunäkymä

## Versionhallinta

Versio	Päiväys	Tehnyt	Muutokset
0.1	7.5.2004	RP, OP	
0.2	22.5.2004	JL, OP	SideKick-pluginin muutokset lisätty, lukuja 3 ja 4 kirjoiteltu.
0.3	23.5.2004	JL, OP	XML-pluginin muutokset lisätty, lukuja 3, 4, 6 ja 7 täydennetty.
0.4	24.5.2004	JL, RP, OP	Muokattu katselmoitavaan tilaan.
1.0	27.5.2004	RP, OP, JR	Katselmoinnissa esille tulleet virheet ja puutteet korjattu.

## Tekijöiden lyhenteet

- JL:** Jani Lirkki  
**OP:** Olavi Paananen  
**RP:** Raimo Pitkänen  
**JR:** Jussi Räisänen

# Sisältö

<b>1</b>	<b>Johdanto</b>	<b>1</b>
<b>2</b>	<b>Termejä</b>	<b>2</b>
2.1	Aihepiirin yleisiä termejä . . . . .	2
2.2	Sovellukseen liittyvät käsitteet . . . . .	5
<b>3</b>	<b>Toteutetut ominaisuudet</b>	<b>6</b>
3.1	jEdit-editorin rakenne . . . . .	6
3.2	Plug-init . . . . .	6
3.3	Xetor-projektin toteuttamat ominaisuudet . . . . .	7
3.3.1	Puunäkymä . . . . .	7
3.3.2	Sisällönmuokkausikkuna . . . . .	7
3.3.3	Elementtilista . . . . .	7
3.3.4	Muut ominaisuudet . . . . .	7
3.4	jEditin ja valmiiden plug-inien toteuttamat ominaisuudet . . . . .	8
3.4.1	Puunäkymä . . . . .	8
3.4.2	Elementtilista . . . . .	8
3.4.3	Tekstinäkymä . . . . .	8
3.4.4	Virhealue . . . . .	8
3.4.5	Esikatselunäkymä . . . . .	8
3.4.6	Ominaisuuseditori . . . . .	9
3.4.7	Muut . . . . .	9
<b>4</b>	<b>Käyttöliittymä</b>	<b>10</b>
4.1	Rakenne . . . . .	10
4.2	Puunäkymä ja elementtilista . . . . .	11
4.2.1	Puun osan valitseminen . . . . .	11
4.2.2	Leikepöydälle kopiointi . . . . .	12
4.2.3	Leikepöydältä liittäminen . . . . .	12
4.2.4	Elementin poistaminen . . . . .	12
4.2.5	Tekstin lisääminen elementin sisälle . . . . .	12
4.2.6	Elementin tekstisisällön muokkaaminen . . . . .	12
4.2.7	Elementin lisääminen tekstin sisälle . . . . .	13
4.2.8	Elementin lisääminen . . . . .	13

4.3	Tekstinäkymä . . . . .	14
4.4	Virheikkuna . . . . .	14
4.5	Esikatselunäkymä . . . . .	14
4.6	Ominaisuuseditori . . . . .	14
4.7	Tilapalkki . . . . .	15
<b>5</b>	<b>Toteutus</b>	<b>16</b>
5.1	XetorSidekick . . . . .	16
5.1.1	SideKickParsedData.java . . . . .	16
5.1.2	SideKickTree.java . . . . .	17
5.1.3	GhostElement.java . . . . .	27
5.1.4	TextElement.java . . . . .	29
5.1.5	UpdateTagListMsg.java . . . . .	29
5.1.6	ShowEditTagDialog . . . . .	30
5.2	XetorXML . . . . .	31
5.2.1	SAXParserImpl.java . . . . .	31
5.2.2	XmlActions.java . . . . .	34
5.2.3	XmlListCellRenderer.java . . . . .	35
5.2.4	XmlInsert.java . . . . .	36
5.2.5	GeneralOptionPane.java . . . . .	40
5.3	Muut muutokset . . . . .	41
5.3.1	Parserin vaihto . . . . .	41
5.3.2	Lisätyt DTD:t . . . . .	41
5.3.3	jEdit . . . . .	41
5.3.4	Property-tiedostot . . . . .	41
5.3.5	Options . . . . .	42
<b>6</b>	<b>Sovelluksen virheet</b>	<b>43</b>
6.1	Puunäkymä . . . . .	43
6.2	Elementtilista . . . . .	43
<b>7</b>	<b>Jatkokehitykseen siirretyt ominaisuudet</b>	<b>44</b>
7.1	CSS-editori . . . . .	44
7.2	Esikatselunäkymä . . . . .	44
7.3	Etsi/korvaa-toiminto . . . . .	44

7.4	jEditin kääntäminen suomen kielelle . . . . .	44
7.5	Muita pienempiä ominaisuuksia . . . . .	45
<b>8</b>	<b>Yhteenveto</b>	<b>46</b>
	<b>Lähteet</b>	<b>47</b>

# 1 Johdanto

Xetor on Jyväskylän yliopiston tietotekniikan sovellusprojekti, jonka tehtävä on suunnitella ja toteuttaa XML-editori Jyväskylän yliopiston Informaatioteknologian tiedekunnalle. Xetor-projekti toteutti määritellyt vaatimukset jEdit-editorin pohjalta, koska se toteutti jo valmiiksi suuren osan Xetor XML-editorille määritellyistä vaatimuksista.

Tämän dokumentin tarkoitus on kuvata Xetor-projektin laajentamat plug-init ja antaa lukijalle yleiskuva jEdit-editorin rakenteesta ja toiminnoista. Lisäksi dokumentissa kuvataan, mitkä määritetyt ominaisuudet saatiin toteutettua ja mitkä ominaisuudet jäävät jatkokehitykseen.

Luvussa 2 esitellään dokumentissa käytettäviä termejä. Luvussa 3 kuvataan sovellukselle asetettujen vaatimusten toteutumista. Luvussa 4 kuvataan jEdit-editorin käyttöliittymää. Xetor-projektin sovellukseen toteuttamista komponenteista ja niiden toiminnasta kerrotaan luvussa 5. Luvussa 6 kerrotaan sovelluksen virheistä ja puutteista ja luvussa 7 listataan jatkokehitykseen jäävät ominaisuudet ja annetaan ideoita jatkokehittäjille. Luku 8 sisältää yhteenvedon dokumentin sisällöstä.

## 2 Termejä

Tässä luvussa on kuvattu projektin aihealueeseen liittyviä termejä. Termit on jaoteltu aihepiirin yleisiin termeihin ja sovellukseen liittyviin käsitteisiin.

### 2.1 Aihepiirin yleisiä termejä

<b>Attribuutti</b>	on elementin ominaisuus, joka tarkoittaa elementin tilaa.
<b>CSS</b>	eli <i>Cascading Style Sheets</i> on XML-dokumenttien ulkoasua kuvaava kieli. [1]
<b>DOM</b>	eli <i>Document Object Model</i> , on alusta- ja kieliriippumaton oliorajapinta, joka antaa skriptien ja ohjelmien päästä käsiksi dokumentin sisältöön, rakenteeseen ja tyyliin. [2]
<b>DTD</b>	on XML-kielissä käytettävä dokumenttityypin määrittelytiedosto. [8]
<b>Elementti</b>	on osa, josta XML-dokumentit koostuvat. Elementit voidaan määrittellä DTD:ssä.
<b>HTML</b>	eli <i>Hypertext Markup Language</i> on standardi merkintäkieli, jolla kuvaillaan www-sivujen sisällön rakenne. [3]
<b>Hyvin muodostettu</b>	on käsite, jolla kuvataan XML-dokumentin rakennetta. Dokumenttia kutsutaan hyvin muodostetuksi, jos sen rakenne on XML-spesifikaation minimisääntöjen mukainen. [8]
<b>ISO-8859-1</b>	on merkitö, joka sisältää ASCII-merkistöön kuuluvien normaalien aakkosten, numeroiden ja yleisimpien välimerkkien lisäksi useimmat länsi- ja pohjoiseurooppalaisten kielten tarvitsemat aksentoidut merkit.
<b>ISO-8859-15</b>	on ISO-8859-standardiin kuuluva merkitö, joka on tarkoitettu käyttöön Euroopan alueelle. Se on muokattu ISO-8859-1:n pohjalta ja siihen on esimerkiksi lisätty euro-merkki.
<b>Java</b>	on Sunin kehittämä laitteistoriippumaton olio-ohjelmointikieli. [4]
<b>JAXP</b>	eli <i>Java API for XML Processing</i> on XML-jäsentimien käytön mahdollistava yhtenäinen rajapinta. [4]



<b>JDK</b>	eli <i>Java Development Kit</i> on Java-ohjelmien standardi kehitysympäristö Sun Microsystemsiltä. [4]
<b>JRE</b>	eli <i>Java Runtime Environment</i> on apuohjelmisto, joka tarvitaan Java-ohjelmien ajamiseen. JRE pitää sisällään mm. ympäristöön sopivan Java-virtuaalikoneen. [4]
<b>Käyttöjärjestelmä</b>	on ohjelmisto, joka ohjaa tietokonetta ja siihen kytkettyjä oheislaitteita.
<b>Käyttötapaus</b>	(engl. <i>use case</i> ) on käyttäjän tai sovelluksen toimintoa tietyn tehtävän suorittamiseksi kuvaava dokumentti.
<b>Linux</b>	on suosittu käyttöjärjestelmä.
<b>Mozilla</b>	on tehokas ja ilmainen standardien mukainen WWW-selain, jonka lähdekoodi on vapaasti käytettävissä. [?]
<b>Parseri</b>	eli jäsennin on ohjelmisto, joka suorittaa tarvittavan sanallisen ja syntaksisen analyysin dokumentille sen rakenteen tunnistamiseksi.
<b>Plug-in</b>	eli täydennysohjelma on sovelluksen ulkopuolinen ohjelma, joka liitetään pääohjelmaan täydentämään alkuperäisen ohjelman toimintoja.
<b>PNG</b>	<i>Portable Network Graphics</i> on W3-yhteenliittymän (W3C) vuonna 1996 määrittelemä suositus tiedostomuodoksi häviöttömälle, siirrettävälle ja tehokkaasti pakatulle rasterikuvalle.
<b>Prosessointiohje</b>	on XML-dokumenttiin liitetty komento tai ohje, jonka XML-jäsennin välittää dokumenttia käsittelevälle sovellukselle. [9]
<b>Selain</b>	on toiminto tai ohjelmisto, jolla selataan tietokantaa. Internet-verkossa asiakasohjelmisto, joilla selataan www-palvelimen sivuja. [5]
<b>UTF-8</b>	on vaihtelevan pituinen merkistön koodaustapa. Merkkikoodista riippuen yksi merkki vie tallennettuna yhdestä neljään tavua.
<b>Validi</b>	on käsite, jolla kuvataan XML-dokumentin DTD:n mukaisuutta. Dokumenttia kutsutaan validiksi, jos se on muodostettu DTD:n mukaisesti.
<b>Windows</b>	on laajaan käyttöön levinnyt käyttöjärjestelmä Microsoftilta.
<b>XHTML</b>	eli <i>Extensible Hypertext Markup Language</i> on XML-muotoinen WWW-dokumenttien kuvaukseen käytettävä kieli. [7]

<b>XML</b>	eli <i>Extensible Markup Language</i> on metakieli, jolla määritellään rakenteellisia merkkaukieliä. [8]
<b>XPath</b>	on yksinkertainen kyselykieli, jolla voidaan hakea erilaisia tietoja XML-dokumentin sisältä.
<b>XSLT</b>	eli <i>Extensible Stylesheet Language Transformations</i> on XML-pohjainen kieli, jonka avulla voidaan XML-dokumentti muuntaa toiseen XML-dokumentin muotoon tai toiseen formaattiin. [10]

## 2.2 Sovellukseen liittyvät käsitteet

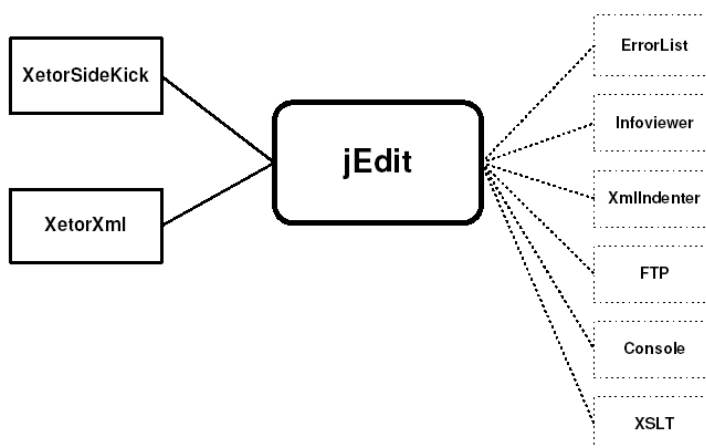
<b>CSS-ominaisuus</b>	on CSS-tiedoston määre, jonka avulla voidaan määrittää XML-dokumentin ulkoasua. [1]
<b>CSS-valitsin</b>	sitoo CSS-ominaisuudet XML-dokumentin elementteihin. [1]
<b>Dialogi</b>	on ikkuna, jonka avulla käyttöliittymän ja käyttäjän välinen kommunikointi tapahtuu.
<b>Esikatselunäkymä</b>	näyttää XML-dokumentin ulkoasun tyylimäärityksineen.
<b>Haamuelementti</b>	on elementti, joka ilmaisee puunäkymässä paikan johon voi lisätä tekstiä tai elementin.
<b>Leikepöytä</b>	graafisten käyttöliittymien ominaisuus, jossa tietoja (tekstiä ja grafiikkaa) voidaan viedä edelleen käsittelyä varten erilliseen käyttöliittymästä varattuun muistiin (leikepöytään) ja tarvittaessa poimia sieltä. [5]
<b>Näkymäikkuna</b>	on ikkuna, joka sisältää puu-, teksti- ja esikatselunäkymän dokumenttiin.
<b>Ominaisuuseditori</b>	( <i>Element inspector</i> )näyttää XML-elementin attribuutit ja niiden arvot. Ominaisuuseditorin kautta voi myös muokata elementin CSS-tyylimäärityksiä.
<b>Puunäkymä</b>	( <i>Tree view</i> ) on hierarkkinen näkymä, jossa voidaan havainnollisesti muokata XML-dokumentin rakennetta.
<b>Tekstidokumentti</b>	eli tekstitiedosto on tiedosto, jossa on tekstiä (kirjaimia, numeroita ja symboleita), mutta ei muotoilukoodeja. Se voi olla ASCII-tiedosto, jonka useimmat tietokoneet pystyvät lukemaan. [5]
<b>Tekstinäkymä</b>	( <i>Text view</i> ) näyttää dokumentin tekstisisällön.
<b>Validointi</b>	on operaatio, jossa tarkistetaan, että XML-dokumentti on hyvinmuodostettu ja validi.
<b>Virhealue</b>	on rajattu alue tekstinäkymästä, jossa virheilmoitukset esitetään validoinnin epäonnistuessa.
<b>XML-dokumentti</b>	on dokumentti, jossa tieto esitetään XML-muodossa. [8]

### 3 Toteutetut ominaisuudet

Tässä luvussa listataan kerrotaan hieman jEdit-editorin rakenteesta ja listataan ominaisuudet, jotka Xetor-projekti on toteuttanut laajentamalla SideKick- ja XML- plug-ineja. Lisäksi lopussa on vielä listattu määritettyjä ominaisuuksia, jotka jEdit-editori toteuttaa valmiiksi. Vaatimusmäärittelyssä [11] on listattu tarkasti kaikki sovellukselle asetetut vaatimukset, joten tässä luvussa listataan sovellukseen toteutetut ominaisuudet hieman yleisemmällä tasolla.

#### 3.1 jEdit-editorin rakenne

jEdit-editori koostuu itse jEdit-editorista ja siihen mahdollisesti asennetuista plug-ineista. Xetor-projekti laajensi SideKick- ja XML-plug-inia. Lisäksi valmiit plug-init, jotka sisällytetään tilaajille annettavaan pakettiin ovat ErrorList, InfoViewer, XMLIndenter, FTP, Console ja XSLT. Tämä on esitetty kuvassa 3.1.



Kuva 3.1: jEditin rakenne.

#### 3.2 Plug-init

Plug-in on jEdit-editoriin saatava laajennos tai lisäosa, jonka avulla saa ohjelmaan lisätoimintoja. Xetor-projekti ei toteuttanut yhtään omaa plug-inia vaan laajensi SideKick- ja XML-plug-ineja. Lisätietoja oman plug-inin rakentamisesta löytyy jEditin käyttöohjeista [13].

### **3.3 Xetor-projektin toteuttamat ominaisuudet**

Tässä luvussa on listattu elementtilistalle, puunäkymälle ja sen yhteydessä olevalle sisällönmuokkausikkunalle toteutetut ominaisuudet. Lisäksi on listattu muut toteutetut ominaisuudet. Xetor-projektin laajentamat SideKick- ja XML-plug-in toteuttavat kyseiset ominaisuudet yhdessä.

#### **3.3.1 Puunäkymä**

- Puunäkymästä voi poistaa, kopioida ja leikata elementtejä
- Puunäkymään voi liittää hyvinmuodostettua XML:ää
- Puunäkymän voi tallentaa PNG-kuvana
- Jos dokumentissa on virheitä, puu lukittuu

#### **3.3.2 Sisällönmuokkausikkuna**

- Sisällönmuokkausikkuna näyttää elementin tekstisisällön
- Sisällönmuokkausikkunan kautta voi lisätä elementtejä tekstin sisälle
- XML-dokumentin tekstisisältöä voi muokata sisällönmuokkausikkunan avulla

#### **3.3.3 Elementtilista**

- XML-dokumenttiin voi lisätä elementtejä elementtilistan avulla
- Elementtilista ehdottaa DTD:n mukaiset lisäykset puusta valittuun kohtaan
- Jos muokattavana on hyvinmuodostettu dokumentti, elementtilista näyttää kaikki dokumentista löytyvät elementit tai elementin nimen voi syöttää itse

#### **3.3.4 Muut ominaisuudet**

- Sovellukseen on lisätty mahdollisuus vaihtaa käyttöliittymä suomenkieliseksi tai englanninkieliseksi

## **3.4 jEditin ja valmiiden plug-inien toteuttamat ominaisuudet**

Tässä luvussa listataan sovellukselta vaaditut ominaisuudet, jotka jEdit tai sen valmiit plug-init toteuttaa.

### **3.4.1 Puunäkymä**

- Puunäkymä näyttää XML-dokumentin hierarkkisena rakenteena

### **3.4.2 Elementtilista**

- Elementtilista näyttää DTD:n mukaiset elementit kohtaan, jossa tekstinäkymän kursori sijaitsee

### **3.4.3 Tekstinäkymä**

- Tekstinäkymä näyttää XML-syntaksin eri osat eri väreillä
- Dokumentti validoidaan, kun valittu aika viimeisestä lisäyksestä kuluu umpeen
- Tekstinäkymä ehdottaa DTD:n mukaiset elementit kursorin kohtaan
- Tekstinäkymässä on automaattinen rivitys
- Tekstinäkymä näyttää rivinumerot haluttaessa
- Tekstinäkymän kirjasinta ja sen väriä ja kokoa pystyy muuttamaan

### **3.4.4 Virhealue**

- Virhenäkymä näyttää dokumentin virheet
- Virheilmoituksesta on linkki tekstinäkymän virheelliselle riville
- Virheilmoituksesta näkyy virheen rivinnumero

### **3.4.5 Esikatselunäkymä**

- Näyttää muokattavan dokumentin ulkoisessa selaimessa

### **3.4.6 Ominaisuuseditori**

- Ominaisuuseditorin kautta voi katsoa ja muokata elementin attribuuttien arvoja

### **3.4.7 Muut**

- Sovellus tallentaa käyttäjän tekemät asetukset XML-muodossa
- Sovellus tallentaa käytetyt DTD:t lokaalille koneelle
- Sovellus sisältää kumoa- ja toista- toiminnot
- Sovellus tallentaa käyttöliittymän tilan
- Sovelluksella pystyy tekemään XSL-muunnoksia
- Sovelluksella on mahdollista muokata dokumentteja suoraan palvelimelta
- Käsiteltävä dokumentti voidaan syöttää ulkoiselle ohjelmalle

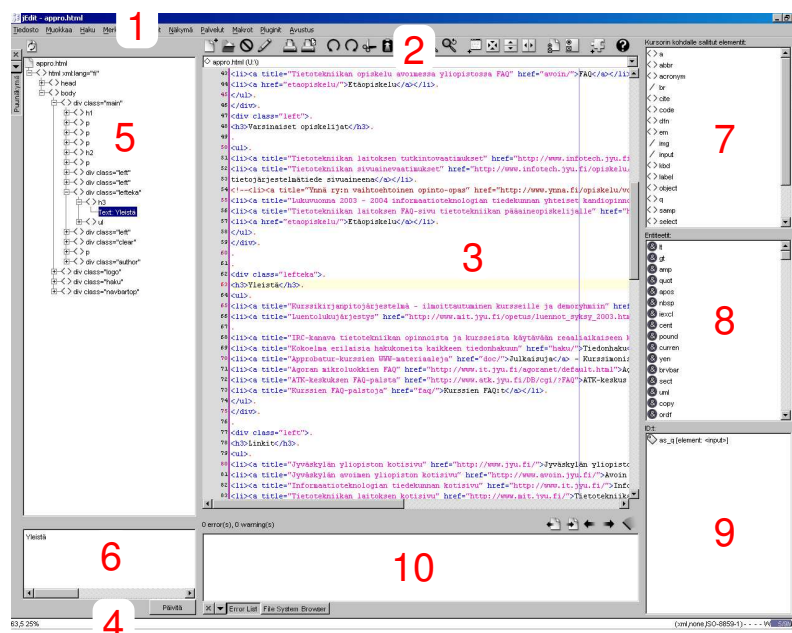
## 4 Käyttöliittymä

Tässä luvussa kuvataan käyttöliittymän rakennetta ja toimintaa. jEdit-editorin käyttöliittymän toimintoihin luodaan vain yleiskuva, mutta Xetor-plug-inien osalta toiminnot kuvataan yksityiskohtaisesti.

### 4.1 Rakenne

Kuvasta 4.2 näkyy jEdit-editorin käyttöliittymän rakenne, kun siihen on asennettu Xetor-projektin laajentamat plug-init. Käyttöliittymän osat on numeroitu ja seuraavassa luvussa kuvataan eri osien toimintaa.

jEditin käyttöliittymä koostuu seuraavista osista: valikkorivi(1), työkalurivi(2), tekstinäkymä(3) ja tilapalkki(4). Laajennettujen plug-inien käyttöliittymä koostuu seuraavista osista: puunäkymä(5), sisällönmuokkausikkuna(6), elementtilista(7), entiteettilista(8) ja tunnistelista(9). Lisäksi käyttöliittymään kuuluu virhealue(10), jonka toteuttaa erillinen `ErrorList`-plug-in.

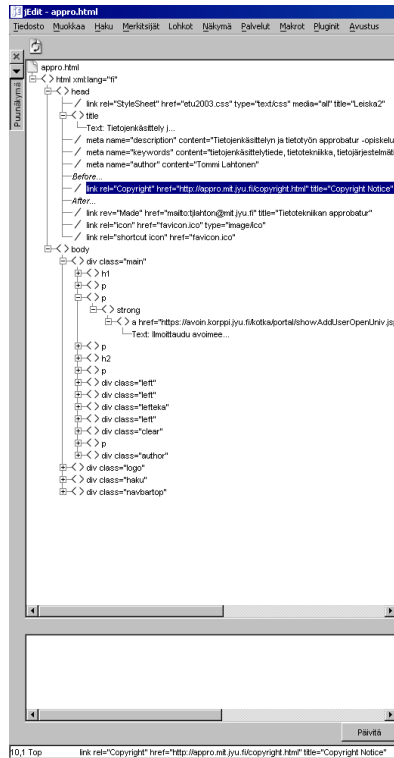


Kuva 4.2: Käyttöliittymän rakenne.



## 4.2 Puunäkymä ja elementtilista

Puunäkymässä XML-dokumentin rakenne näytetään hierarkkisen rakenteena. Puun solmut voivat olla joko XML-elementtejä tai tekstiä. Tekstistä näytetään puunäkymässä vain alku, mutta koko sisältöä voi tarkastella ja muokata sisällönmuokkausikkunassa. Elementin sisällä olevat elementit näytetään lapsisolmuina. Validoinnin epäonnistuessa puunäkymä lukittuu ja sitä voi muokata vasta, kun virheet on korjattu tekstinäkymän kautta.



Kuva 4.3: Puunäkymä.

### 4.2.1 Puun osan valitseminen

Puun osan voi valita sekä hiiren vasemmalla että oikealla painikkeella. Vasemmalla painikkeella valitaan vain puun osa ja oikealla painikkeella avataan lisäksi kontekstivalikko. Puun osia valittaessa voidaan käyttää myös **Shift**- ja **Ctrl**-näppäimiä. Valittaessa **Shift**-näppäin pohjassa puusta elementti maalautuu kyseinen elementti myös tekstinäkymässä. Valittaessa **Ctrl**-näppäin pohjassa puusta elementti näytetään tekstinäkymässä vain kyseisen elementin sisältö. Kun puusta on valittu elementti, ilmestyy sen ennen, jälkeen ja sisään haamuelementti. Valitsemalla haamuelementin ja elementin lisäyksen ilmestyy kyseiseen kohtaan haluttu elementti.

#### **4.2.2 Leikepöydälle kopiointi**

Käyttäjä valitsee puusta kopioitavan elementin ja tämän jälkeen kopiointi tapahtuu valitsemalla hiiren oikealla painikkeella avautuvasta kontekstivalikosta kohta `Kopioi`.

#### **4.2.3 Leikepöydältä liittäminen**

Käyttäjä valitsee hiiren oikealla painikkeella puusta kohdan, johon haluaa lisätä leikepöydän sisällön. Ruudulle avautuu kontekstivalikko, josta käyttäjä valitsee `Liitä`-toiminnon. Sen jälkeen hän valitsee kontekstivalikosta jonkin toiminnoista `Liitä` ennen, `Liitä sisään` tai `Liitä jälkeen`. Mikäli liitettävä teksti ei ole hyvin muodostettua, liittämistä ei suoriteta. Muussa tapauksessa teksti liitetään valittuun kohtaan.

#### **4.2.4 Elementin poistaminen**

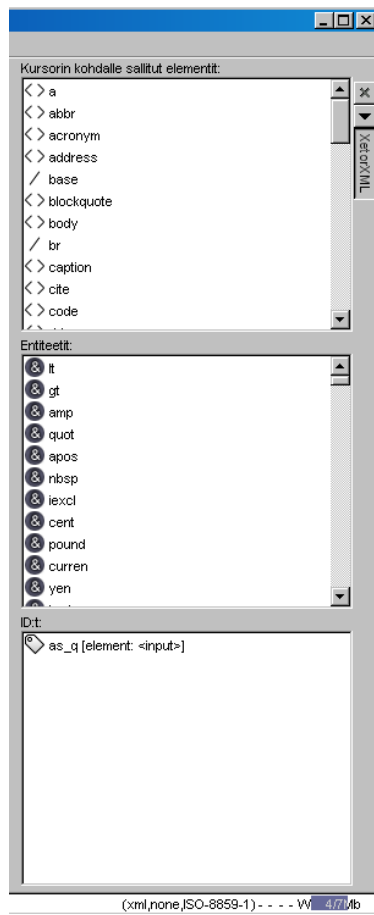
Käyttäjä voi poistaa elementin hiirellä valitsemalla ensin poistettavan elementin hiiren oikealla painikkeella ja valitsemalla avautuvasta kontekstivalikosta kohdan `Poista`. Valitun elementin voi poistaa myös painamalla `Delete`-näppäintä.

#### **4.2.5 Tekstin lisääminen elementin sisälle**

Käyttäjä valitsee puusta haamuelementin siitä kohdasta johon haluaa lisätä tekstiä. Sen jälkeen käyttäjä kirjoittaa haluamansa tekstin sisällönmuokkausikkunaan ja siirtää kohdistimen pois sisällönmuokkausikkunasta, jolloin teksti päivittyy dokumenttiin. Kohdistimen voi siirtää pois sisällönmuokkausikkunasta esimerkiksi painamalla lisää-nappulaa, joka on lisätty tarjoamaan lisäämiselle jokin intuitiivisesti luonteva toimintamalli.

#### **4.2.6 Elementin tekstisisällön muokkaaminen**

Käyttäjä valitsee puusta sen tekstielementin, jota haluaa muokata. Valinnan jälkeen teksti ilmestyy sisällönmuokkausikkunaan. Käyttäjä muuttaa tekstiä sisällönmuokkausikkunassa ja siirtää kohdistimen pois, jolloin teksti päivittyy dokumenttiin.



Kuva 4.4: Elementti-, entiteetti- ja tunnistelista.

#### 4.2.7 Elementin lisääminen tekstin sisälle

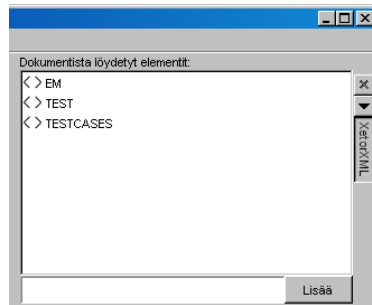
Käyttäjä valitsee puusta tekstielementin jonka jälkeen hän valitsee lisättävän elementin paikan sisällönmuokkausikkunasta siirtämällä kursorin haluttuun paikkaan tai maalaamalla tekstin, jonka ympärille haluaa lisätä elementin. Tämän jälkeen hän suorittaa elementin lisäyksen elementtilistan avulla(4.2.8).

#### 4.2.8 Elementin lisääminen

Käyttäjä valitsee ensin puusta joko haamuelementin kohdasta, johon haluaa lisätä elementin tai XML-elementin, jonka haluaa ympäröitävän elementillä. Sen jälkeen hän valitsee elementtilistasta lisättävän elementin. Jos elementti valitaan hiiren oikealla painikkeella, se lisätään heti dokumenttiin ja sen pakolliset attribuutit asetetaan oletusarvoonsa tai tyhjäksi. Mikäli valinta tehdään hiiren vasemmalla painikkeella näytetään käyttäjälle ominaisuusedi-

tori.

Jos muokattava dokumentti on pelkästään hyvinmuodostettu voi käyttäjä myös syöttää lisäävän elementin nimen elementtilistan alapuolella olevaan tekstikenttään. Tämän jälkeen elementin lisääminen tapahtuu painamalla tekstikentän vieressä olevaa lisää-nappia.



Kuva 4.5: Elementtilista hyvinmuodostettua XML-dokumenttia käsiteltäessä.

### 4.3 Tekstinäkymä

Tekstinäkymä esittää XML-dokumentin tekstimuotoisena. Xetor-ryhmä ei ole tekstinäkymän toiminnallisuutta muokannut, joten sen toimintaa ei käsitellä enempää tässä dokumentissa.

### 4.4 Virheikkuna

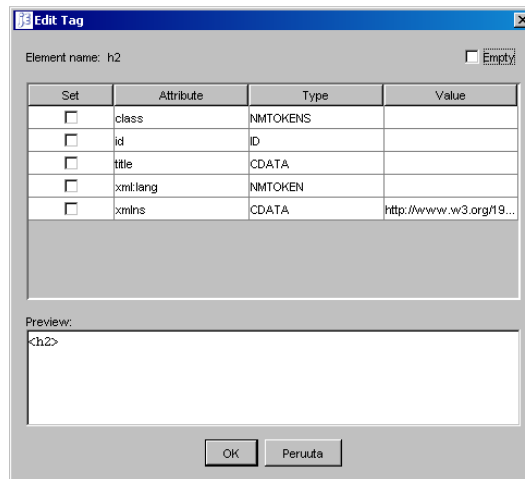
Virheikkuna esittää dokumentin validoinnissa esille tulleet virheet. Virheikkunan toiminnallisuuden toteuttaa `ERRORList`-plug-in. Xetor-ryhmä ei ole virheikkunan toiminnallisuutta muokannut, joten asiaa ei käsitellä enempää tässä dokumentissa.

### 4.5 Esikatselunäkymä

Esikatselunäkymä näyttää dokumentin tyylimäärityksineen. Toiminnallisuus on toteutettu `InfoViewer`-plug-inissa, joten sitä ei raportoida tarkemmin tässä dokumentissa.

### 4.6 Ominaisuuseditori

Yksi tärkeä käyttöliittymän osa, jota ei mainittu on ominaisuuseditori. Ominaisuuseditori saadaan halutessa näkyviin, kun lisätään elementtiä. Ominaisuuseditorilla voi muokata elementin attribuuttimäärityksiä.



Kuva 4.6: Ominaisuuseditori.

## 4.7 Tilapalkki

Tilapalkissa näytetään lyhyitä tekstimuotoisia ilmoituksia suoritetuista toiminnoista.

## 5 Toteutus

Tässä luvussa esitetään Xetor plug-inien luokat ja niiden metodien toiminnallisuus. Java-dokumentaatio on nähtävillä osoitteessa

<http://sovellusprojektit.it.jyu.fi/xetor/ohjelmakoodi/source/javadoc>

### 5.1 XetorSidekick

Xetor projekti toteutti puunäkymän png-kuvaksi tallentamisen Erkka-projektilta kopioitujen tiedostojen

- `ExampleFileFilter.java`,
- `PngEncoder.java` ja
- `PngEncoderB.java`

avulla. Nämä luokat on lisätty `sidekick`-pakettiin, mutta näiden luokkien toiminnallisuutta ei käsitellä tässä dokumentissa.

#### 5.1.1 SideKickParsedData.java

##### Muokatut metodit:

```
getTreePathForPosition
```

**Kuvaus:** Kolmiparametrisella `getTreePathForPosition`-metodilla voidaan tarkistaa, onko tekstialueen kursori parametrina tuotavan solmun tai jonkin sen lapsisolmun sisällä. Xetor on lisännyt tähän tapauksen, jossa puun solmu on `TextElement`-olio.

**Parametrit:** `TreeNode node` - puun solmu, jota ollaan tutkimassa  
`int dot` - offset tekstipuskurissa  
`List path` - Lista, johon tallennetaan ne solmut jotka ovat kursorille vievällä polulla

**Palautusarvo:** `boolean` - onko kursori parametrina tuodun puun solmun sisällä

## 5.1.2 SideKickTree.java

### Lisätyt yksityiset attribuutit:

- JEditorPane contentArea
- JPanel changeTextButtonPanel
- JButton changeTextButton
- JFileChooser fc
- File file
- JScrollPane treepanel

### Lisätyt julkiset attribuutit:

Näitä julkisia staattisia attribuutteja käytetään `paste`-metodin parametrina kertomaan mihin liittäminen halutaan suorittaa valittuun elementtiin nähden.

- BEFORE = 0
- INSIDE = 1
- AFTER = 2

### Lisätyt metodit:

```
getTree
```

**Kuvaus:** Palauttaa luokan sisäänsä kapseloiman puun.

**Parametrit:** –

**Palautusarvo:** CustomTree - Puu

```
treeHasFocus
```

**Kuvaus:** Kertoo, onko puulla focus.

**Parametrit:** –

**Palautusarvo:** boolean - Onko puulla focus

`contentAreaHasFocus`

**Kuvaus:** Kertoo, onko sisällönmuokkausikkunalla focus.

**Parametrit:**

**Palautusarvo:** boolean - onko sisällönmuokkausikkunalla focus

`activateTree`

**Kuvaus:** Aktivoi puun. Tätä metodia kutsutaan parsinnan jälkeen, jos XML-dokumentissa ei ole virheitä.

**Parametrit:** –

**Palautusarvo:** –

`deActivateTree`

**Kuvaus:** Poistaa puun aktiivisuuden. Tätä metodia kutsutaan kun XML-dokumentissa on virheitä.

**Parametrit:** –

**Palautusarvo:** –

`cut`

**Kuvaus:** Kopioi valitun elementin leikepöydälle XML-muotoisena ja poistaa elementin dokumentista ja puusta.

**Parametrit:** –

**Palautusarvo:** –

`copy`

**Kuvaus:** Kopioi valitun elementin leikepöydälle XML-muotoisena.

**Parametrit:** –

**Palautusarvo:** –



paste

**Kuvaus:** Liittää leikepöydän sisällön joko valitun elementin sisälle, elementtiä ennen tai elementin jälkeen riippuen where-parametrasta.

**Parametrit:** –

**Palautusarvo:** int where - pitää olla jokin seuraavista:  
SideKickTree.BEFORE, SideKickTree.INSIDE,  
SideKickTree.AFTER

delete

**Kuvaus:** Poistaa puussa valitun solmun puusta ja dokumentista.

**Parametrit:** –

**Palautusarvo:** –

getSelectedElement

**Kuvaus:** Palauttaa puussa valitun elementin, mikäli se on tyyppiä GhostElement, muussa tapauksessa palauttaa null.

**Parametrit:** –

**Palautusarvo:** GhostElement - Puusta valittu GhostElement tai null jos puussa valittu olio ei ole GhostElement-tyyppiä

deleteSelection

**Kuvaus:** Pyyhkii tekstinäkömästä puussa valitun elementin, mikäli se on Asset-tyyppinen.

**Parametrit:** –

**Palautusarvo:** –

export

**Kuvaus:** Avaa dialogin, johon käyttäjän tulee syöttää tiedoston nimi. Tämän jälkeen tallentaa kutsuhetkellä näkyvän osan puunäkymästä PNG-muotoisena dialogissa määriteltyyn tiedostoon.

**Parametrit:** –

**Palautusarvo:** –

### Muokatut sisäluokat:

## CustomTree

Attribuutit:

- JPopupMenu popup - puunäkymän pop-up-valikko

### Muokatut metodit:

CustomTree

**Kuvaus:** Muodostin, joka kutsuu ylikuokan muodostinta. Xetor on lisännyt tänne hiiren oikean napin painalluksella avautuvan pop-up-valikon luomisen jEditin GUIUtilities-luokan avulla.

**Parametrit:** TreeModel model - puun sisältö

**Palautusarvo:** –

`processMouseEvent`

**Kuvaus:** Component-yliluokan ylikirjoitettu hiiritapahtumien käsittelijä. Xetor on lisännyt tänne hiiren oikealla napilla tapahtuvan pop-up-valikon näyttämisen. Pop-up-valikko näytetään jEditin GUIUtilities-luokan avulla.

**Parametrit:** MouseEvent evt

**Palautusarvo:** -

#### Lisätyt metodit:

`pasteFromClipboard`

**Kuvaus:** Parsii leikepöydän sisällön ja mikäli se on hyvin muodostettua XML:ää, se liitetään dokumenttiin ennen puussa valittua elementtiä tai elementin jälkeen.

**Parametrit:** boolean before - liitetäänkö ennen elementtiä vai elementin jälkeen

**Palautusarvo:** -

`copySelectionToClipboard`

**Kuvaus:** Kopioi puussa valitun elementin leikepöydälle XML-muotoisena.

**Parametrit:** -

**Palautusarvo:** -

`hasRoot`

**Kuvaus:** Palauttaa true jos XML-dokumentin juurielementti löytyy puusta ja false mikäli puussa ei ole juurielementtiä.

**Parametrit:** -

**Palautusarvo:** boolean - onko XML-dokumentilla juurielementtiä

`isRootSelected`

**Kuvaus:** Palauttaa `true` jos XML-dokumentin juurielementti on valittuna ja `false` mikäli juurielementti ei ole valittuna.

**Parametrit:** –

**Palautusarvo:** `boolean` - onko XML-dokumentin juurielementti valittuna

`getCurrentAsset`

**Kuvaus:** Palauttaa puussa valitun solmun, mikäli se on `Asset`-olio. Mikäli se ei ole `Asset`-olio, palautetaan `null`.

**Parametrit:** –

**Palautusarvo:** `Asset` - puussa valittu solmu

`pasteInside`

**Kuvaus:** Parsii leikepöydän sisällön ja mikäli se on hyvin muodostettua XML:ää, se liitetään dokumenttiin puussa valitun elementin sisään (alkuun).

**Parametrit:** –

**Palautusarvo:** –

## KeyHandler

### Muokatut metodit:

`keyPressed`

**Kuvaus:** KeyAdapter-luokan ylikirjoitettu metodi, jota kutsutaan aina kun näppäimistöltä on painettu jotain näppäintä. Xetor on lisännyt metodiin delete-näppäimen painalluksella suoritettavan puussa valitun solmun poistamisen.

**Parametrit:** KeyEvent evt

**Palautusarvo:** –

## Renderer

### Muokatut metodit:

`getTreeCellRendererComponent`

**Kuvaus:** Ylikirjoitettu DefaultTreeCellRenderer-yliluokan metodi, jolla asetetaan puun piirtokomponentti oikeaan tilaan. Metodia kutsutaan kullekin puun solmulle erikseen ja kutsun jälkeen piirto suoritetaan määritysten mukaisesti. Xetor on lisännyt tähän metodiin haamuelementtien fontin vaihdon kursivoiduksi.

**Parametrit:** JTree tree  
Object value  
boolean sel  
boolean expanded  
boolean leaf  
int row  
boolean hasFocus

**Palautusarvo:** Component

## Lisätyt sisäluokat:

### **ContentAreaFocusAdapter**

Yksityiset attribuutit:

- CustomTree tree - osoitin puuhun
- JEditorPane contentArea - osoitin sisällönmuokkausikkunaan

ContentAreaFocusAdapter

**Kuvaus:** Muodostin, joka asettaa luokan puu- ja editorpane-attribuutteihin parametrinä tuodun puun ja editorpanen.

**Parametrit:** CustomTree tree  
JEditorPane contentArea

**Palautusarvo:** -

focusGained

**Kuvaus:** FocusAdapter-luokan ylikirjoitettu metodi, joka pitää huolen siitä, että sisällönmuokkausikkunan kursori on oikeassa paikassa kun se saa focusen.

**Parametrit:** FocusEvent e

**Palautusarvo:** -

focusLost

**Kuvaus:** FocusAdapter-luokan ylikirjoitettu metodi, joka lisää sisällönmuokkausikkunan tekstin tekstinäkymään. Mikäli puussa valittu elementti on haamuelementti niin teksti lisätään kyseiseen kohtaan dokumentissa. Mikäli puussa valittu elementti on tekstielementti, korvataan tekstielementin tekstisisältö sisällönmuokkausikkunan tekstillä.

**Parametrit:** FocusEvent e

**Palautusarvo:** -

## ContentAreaCaretListener

Yksityiset attribuutit:

- CustomTree tree - osoitin puuhun
- JEditorPane contentArea - osoitin sisällönmuokkausikkunaan

ContentAreaCaretListener

**Kuvaus:** Muodostin, joka asettaa luokan puu- ja editorpane-attribuutteihin parametrinä tuodun puun ja editorpanen.

**Parametrit:** CustomTree tree  
JEditorPane contentArea

**Palautusarvo:** -

caretUpdate

**Kuvaus:** CaretListener-rajapinnan toteutettu metodi. Jos puussa on valittu TextElement, niin lasketaan valinnan alun ja lopun offset sekä lähetetään EditBusiin UpdateTagListMsg-viesti, jotta elementtilista päivittyisi.

**Parametrit:** CaretEvent e

**Palautusarvo:** -

## MyHandler

MyHandler on tarkoitettu vain tyhmäksi SAX-parsinnassa tapahtuneiden poikkeusten eteenpäin heittäjäksi. Tätä käytetään parsittaessa leikepöydältä dokumenttiin liitettävää XML:ää.

MyHandler

**Kuvaus:** Muodostin jossa ainoastaan kutsutaan `org.xml.sax.helpers.DefaultHandler`-yliluokan muodostinta.

**Parametrit:** –

**Palautusarvo:** –

error

**Kuvaus:** Heittää edelleen virheestä saadun poikkeuksen.

**Parametrit:** `SAXParseException e`

**Palautusarvo:** –

warning

**Kuvaus:** Heittää edelleen varoituksesta saadun poikkeuksen.

**Parametrit:** `SAXParseException e`

**Palautusarvo:** –

## MyTreeSelectionListener

Attribuutit:

- `JTree aTree` - osoitin puuhun
- `JEditorPane aEditorPane` - osoitin sisällönmuokkausikkunaan



## MyTreeSelectionListener

**Kuvaus:** Muodostin, joka asettaa luokan puu- ja editorpane-attribuutteihin parametrinä tuodun puun ja editorpanen.

**Parametrit:** JTree tree  
JEditorPane epane

**Palautusarvo:** -

## valueChanged

**Kuvaus:** TreeSelectionListener-rajapinnan metodi, jota kutsutaan aina kun valinta puussa on vaihtunut. Poistaa edelliseltä valitulta solmulta GhostElementit ja laittaa uudelle valitulle elementille GhostElementit. Vanhoja GhostElementtejä ei poisteta, jos uusi valittu elementti on GhostElement. Metodi lähettää myös EditBusiin UpdateTagListMsg-viestin, joka käsitellään XmlInsert-luokassa siten, että ehdotettavat elementit päivitetään vastaamaan uutta valintaa. Jos uusi valinta on TextElement, laitetaan teksti sisällönmuokkausikkunaan.

**Parametrit:** TreeSelectionEvent e

**Palautusarvo:** -

### 5.1.3 GhostElement.java

#### Yksityiset attribuutit:

- int aOffsetStart - GhostElementin alun offset tekstipuskurissa
- int aOffsetEnd - GhostElementin lopun offset tekstipuskurissa
- int aType - GhostElementin tyyppi, BEFORE, INSIDE tai AFTER

### Julkiset attribuutit:

Näitä julkisia staattisia attribuutteja käytetään GhostElementiä luotaessa kertomaan minkä tyyppinen GhostElement luodaan. GhostElementin aType-attribuutin arvona tulee olla jokin näistä kolmesta.

- BEFORE = 0
- INSIDE = 1
- AFTER = 2

#### GhostElement

**Kuvaus:** Muodostin, joka asettaa aOffsetStart-, aOffsetEnd- ja aType-attribuutit parametrina tuotuihin arvoihin.

**Parametrit:** int pOffsetStart  
int pOffsetEnd  
int pType

**Palautusarvo:** -

#### getOffsetStart

**Kuvaus:** Saantimetodi aOffsetStart-attribuutille

**Parametrit:** -

**Palautusarvo:** int - aOffsetStart-attribuutin arvo

#### getOffsetEnd

**Kuvaus:** Saantimetodi aOffsetEnd-attribuutille

**Parametrit:** -

**Palautusarvo:** int - aOffsetEnd-attribuutin arvo

toString

**Kuvaus:** Object-luokan ylikirjoitettu metodi, joka palauttaa tekstimuotoisen esityksen GhostElementin tyypistä (onko BEFORE, INSIDE vai AFTER).

**Parametrit:** –

**Palautusarvo:** String - tekstimuotoinen esitys haamuelementin tyypistä

#### 5.1.4 TextElement.java

##### Julkiset attribuutit:

- String text
- int start
- int end

toString

**Kuvaus:** Object-luokan ylikirjoitettu metodi, joka palauttaa tekstimuotoisen esityksen tekstielementistä muodossa "Text: "+ teksti. Jos teksti on yli 20 merkkiä pitkä, tulosmerkkijonoon lisätään ainoastaan 20 ensimmäistä merkkiä.

**Parametrit:** –

**Palautusarvo:** String - Tekstielementin sisältö muodossa "Text"+ teksti

#### 5.1.5 UpdateTagListMsg.java

Elementtilistan päivityspyyntö -viestiä kuvaava luokka, joka on peritty EBMessagesta jotta UpdateTagList-viestejä voitaisiin lähettää EditBusiin.

##### Yksityiset attribuutit:

- int aOffsetStart - alun offset tekstipuskurissa
- int aOffsetEnd - lopun offset tekstipuskurissa
- View view - osoitin jEdit-teksti-ikkunaan

UpdateTagList

**Kuvaus:** Muodostin, joka asettaa `offsetStart`-, `offsetEnd`- ja `view`-attribuutit parametrina tuotuihin arvoihin ja kutsuu `EBMessage`-yliluokan muodostinta parametrilla `null`.

**Parametrit:** `View view`  
`int offsetStart`  
`int offsetEnd`

**Palautusarvo:** –

getView

**Kuvaus:** Saantimetodi `view`-attribuutille

**Parametrit:** –

**Palautusarvo:** `View` – `view`-attribuutin arvo

getOffsetStart

**Kuvaus:** Saantimetodi `offsetStart`-attribuutille

**Parametrit:** –

**Palautusarvo:** `int` – `offsetStart`-attribuutin arvo

getOffsetEnd

**Kuvaus:** Saantimetodi `offsetEnd`-attribuutille

**Parametrit:** –

**Palautusarvo:** `int` – `offsetEnd`-attribuutin arvo

### 5.1.6 ShowEditTagDialog

Ominaisuuseditorin avaamispyyntö -viestiä kuvaava luokka, joka on peritty `EBMessage`sta, jotta `ShowEditTagDialog`-viestejä voitaisiin lähettää `EditBusiin`.

## 5.2 XetorXML

### 5.2.1 SAXParserImpl.java

#### Muokatut metodit:

`parse`

**Kuvaus:** Parsii parametrina tuodun puskurin. Xetor on lisännyt tähän puunäkymän lukituksen poiston ja validoinnin kytkemisen pois jos dokumenttiin ei ole liitetty DTD:tä.

**Parametrit:** `Buffer buffer`  
`DefaultErrorSource errorSource`

**Palautusarvo:** `SideKickParsedData`

#### Muokatut sisäluokat:

### Handler

Seuraavat attribuutit on lisätty `SAXParserImpl`-luokan `Handler`-sisäluokkaan, jotta saadaan entiteetit ja rivinvaihdot esitettyä oikein `TextElement`-luokassa.

- `boolean lastCallChar` - Onko viimeksi kutsuttu `Characters`-metodia. Tämä asetetaan muodostimessa arvoon `false`.
- `StringBuffer textContent` - Tämä luodaan muodostimessa ja luonnin jälkeen se on tyhjä merkkijonopuskuri.
- `int charStartOffset` - Tämä asetetaan muodostimessa arvoon `-1`.
- `int charEndOffset` - Tämä asetetaan muodostimessa arvoon `-1`.

Luokkaan on lisätty lisäksi staattinen kokonaislukuvakio `MAX_TEXT_LENGTH` joka laitetaan arvoon `20`, joka kertoo kuinka monta merkkiä tekstisisällöstä esitetään puunäkymässä ennen kolmea pistettä.

## Muokatut metodit:

### addError

**Kuvaus:** Lisää virhenäkymään virheilmoituksen. Xetor on lisännyt tähän metodin puunäkymän lukitsemisen havaitun virheen yhteydessä.

**Parametrit:** int type  
String uri  
int line  
String message

**Palautusarvo:** –

### characters

**Kuvaus:** Characters-metodia kutsutaan aina kun XML-parseri löytää dokumentista tekstikokonaisuuden, rivinvaihdon tai entiteetin. Xetor on lisännyt tähän metodiin löydetyn tekstikokonaisuuden, entiteetin tai rivinvaihdon lisäämisen textContent-attribuuttiin, mikäli käyttäjä haluaa nähdä tekstisisällön puussa.

**Parametrit:** char ch[]  
int start  
int length

**Palautusarvo:** –

### endDocument

**Kuvaus:** endDocument-metodia kutsutaan, kun dokumentin SAX-parsinta päättyy. Xetor on lisännyt tähän SideKickUpdate-viestin lähettämisen EditBusiin jotta puunäkymä päivitetäisiin.

**Parametrit:** –

**Palautusarvo:** –

### startElement

**Kuvaus:** startElement-metodia kutsutaan, kun elementin SAX-parsinta alkaa. Xetor on lisännyt tähän addText-metodin kutsun ja lastCallChar-attribuutin falseksi asettamisen, mikäli lastCallChar on true. Metodiin on myös lisätty matchAttributes-metodin kutsu, jossa käydään varmistamassa että parserin löytämät attribuutit todella löytyvät tekstipuskurista.

**Parametrit:** -

**Palautusarvo:** -

### endElement

**Kuvaus:** endElement-metodia kutsutaan, kun elementin SAX-parsinta päättyy. Xetor on lisännyt tähän addText-metodin kutsun ja lastCallChar-attribuutin falseksi asettamisen, mikäli lastCallChar on true.

**Parametrit:** -

**Palautusarvo:** -

### Lisätyt metodit:

### matchAttributes

**Kuvaus:** matchAttributes-metodi on tehty korjaamaan SAX-parsinnan yhteydessä havaittu ongelma, jossa parsinnan tuloksena elementille ilmestyi attribuutteja, joita ei tekstissä ole esitelty (esim. head-elementille profile-attribuutti). matchAttributes vertaa parsinnan tuloksena saatuja attribuutteja tekstipuskurista löytyviin attribuutteihin ja mikäli ne eivät täsmää, poistetaan ylimääräiset attribuutit. Funktio palauttaa oikeat attribuutit kutsujalle.

**Parametrit:** Attributes attrs - parserin löytämät attribuutit  
int offset - puskurin paikka josta attribuuttimääritykset alkaa.

**Palautusarvo:** Attributes

```
addText
```

**Kuvaus:** Luo TextElement-olion textContent-attribuutin sisällöstä ja lisää TextElementin puuhun siihen kohtaan missä parseri on dokumenttia lukemassa. TextElementin start-attribuutin arvoksi asetetaan charStartOffset-attribuutin arvo ja end-attribuutin arvoksi asetetaan charEndOffset-attribuutin arvo. textContent-attribuutti asetetaan tyhjäksi ja charEndOffset sekä charStartOffset asetetaan arvoon -1.

**Parametrit:** -

**Palautusarvo:** -

## 5.2.2 XmlActions.java

### Muokatut metodit:

```
showEditTagDialog
```

**Kuvaus:** Staattinen metodi, joka luo tagin muokkaus -ikkunan ja näyttää sen käyttäjälle. Xetor on lisännyt tähän metodiin kaksi parametria - offsetStart ja offsetEnd - sekä tarkistuksen siitä, kumpi on aktiivinen osa puunäkymässä, puu vai sisällönmuokkausikkuna. Näitä tarvitaan, jotta saadaan elementti lisättyä myös sisällönmuokkausikkunaan, mikäli se on aktiivinen.

**Parametrit:** View view  
ElementDecl elementDecl  
int offsetStart  
int offsetEnd

**Palautusarvo:**



### 5.2.3 XmlListCellRenderer.java

#### Muokatut metodit:

```
getListCellRendererComponent
```

**Kuvaus:** Ylikirjoitettu DefaultListCellRenderer-yliluokan metodi, jolla asetetaan listan piirtokomponentti oikeaan tilaan. Metodia kutsutaan kullekin listan alkiolle erikseen ja kutsun jälkeen piirto suoritetaan määritysten mukaisesti. Xetor on lisännyt tähän metodiin NotValidElementDecl-tyyppisten elementtien käsittelyn, jotta listaan piirretään myös hyvin muodostetun dokumentin elementit.

**Parametrit:** JList list  
Object value  
int index  
boolean isSelected  
boolean cellHasFocus

**Palautusarvo:** Component

## 5.2.4 XmlInsert.java

### Lisätyt yksityiset attribuutit:

- JPanel elementPanel - Käyttöliittymän rakentamiseen tarvittava paneeli.
- JLabel JLabelElementPanel - Teksti, joka lukee ehdotettavien elementtien yläpuolella.
- JTextField elementAdd - Elementin lisäys -tekstikenttä. Käytetään silloin kun muokataan hyvin muodostettua mutta ei-validia dokumenttia.
- JPanel JPanelElementAdd - Käyttöliittymän rakentamiseen tarvittava paneeli.
- JButton JButtonAdd - Elementin lisäys -nappula. Käytetään silloin kun muokataan hyvin muodostettua mutta ei-validia dokumenttia.
- int offsetStart - Alustetaan muodostimessa arvoon -1.
- int offsetEnd - Alustetaan muodostimessa arvoon -1.
- MouseHandler aMouseHandler - Elementtilistan hiirenkäsittelijä, joka on siirretty attribuutiksi, jotta se voidaan ottaa pois silloin kun dokumentti ei ole validi.
- KeyHandler aKeyHandler - Elementtilistan näppäimistönkäsittelijä, joka on siirretty attribuutiksi, jotta se voidaan ottaa pois silloin kun dokumentti ei ole validi.

### Muokatut metodit:

XmlInsert

- Kuvaus:** Muodostin. Xetor on lisännyt tähän metodiin hyvin muodostettuun dokumenttiin elementin lisäämisen mahdollistavien tekstikentän ja nappulan luomisen.
- Parametrit:** View view  
boolean sideBySide
- Palautusarvo:** -

### handleMessage

**Kuvaus:** HandleMessagea kutsutaan aina kun EditBusiin tulee viesti. Xetor on muokannut tätä metodia seuraavasti: Aina kun tulee EditPaneUpdate.BUFFER\_CHANGED-viesti, asetetaan offsetStart- ja offsetEnd-attribuutit arvoon -1. Aina kun tulee SideKickUpdate-viesti, kutsutaan enableElementPanel-metodia. Aina kun tulee UpdateTagListMsg-viesti asetetaan offsetStart- ja offsetEnd-attribuutit viestin antamiin arvoihin ja kutsutaan update-metodia.

**Parametrit:** EBMessage msg - viesti

**Palautusarvo:** –

### setDeclaredElements

**Kuvaus:** Asettaa ehdotettavat elementit parametrin mukaisiksi. Xetor on lisännyt tähän NotValidElementDecl-tyyppisten elementtien käsittelyn.

**Parametrit:** List elements - ehdotettavat elementit

**Palautusarvo:** –

### updateTagList

**Kuvaus:** Päivittää ehdotettavien elementtien listan. Xetor on muokannut metodia seuraavasti: Mikäli dokumentti on hyvin muodostettu, mutta siihen ei ole yhdistetty DTD:tä <!DOCTYPE...-määritteellä, ehdotettavien elementtien listaan laitetaan aakkosjärjestyksessä kaikki erilaiset elementit jotka löydetään puusta. Mikäli offsetStart-attribuutti on pienempi kuin 0, ei ehdoteta mitään elementtejä.

**Parametrit:** –

**Palautusarvo:** –

## Lisätyt metodit:

`insertElement`

**Kuvaus:** `insertElement`-metodia kutsutaan, kun on painettu `ei-validia`, hyvin muodostettua dokumenttia muokattaessa näkyvää elementin lisäys -nappia. Tällöin lisätään elementin lisäys -tekstikenttään kirjoitettu elementti tekstialueessa suoritettuna valinnan ympärille ja jos valintaa ei ole, suoritetaan lisäys puunäkymässä aktiivisena olevan `GhostElementin` kohdalle tai puunäkymässä aktiivisena olevan `Assetin` ympärille. Lisäyksen jälkeen elementtien lisäys -ikkuna otetaan pois käytöstä ja se palautetaan käyttöön, mikäli validointi onnistuu.

**Parametrit:** `String pElementName` - Lisättävän elementin nimi

**Palautusarvo:** -

`getelementAddText`

**Kuvaus:** Palauttaa elementin lisäys -tekstilaatikossa olevan tekstin.

**Parametrit:** -

**Palautusarvo:** `String` - elementin lisäys -tekstilaatikossa oleva teksti

`enableElementPanel`

**Kuvaus:** Kytkee `XmlInsert`-ikkunan komponentit käyttöön.

**Parametrit:** -

**Palautusarvo:** -

`disableElementPanel`

**Kuvaus:** Kytkee `XmlInsert`-ikkunan komponentit pois käytöstä.

**Parametrit:** -

**Palautusarvo:** -

getPaths

**Kuvaus:** getPaths lisää list-parametriin parent-parametrin ja TreePath-oliot kaikkiin sen lapsisolmuihin. Tätä kutsutaan updateTagList-metodista.

**Parametrit:** TreePath parent - polku, josta aloitetaan elementtien etsiminen  
List list - lista, johon elementti lisätään

**Palautusarvo:** -

### Muokatut sisäluokat:

## MouseHandler

### Muokatut metodit:

mouseClicked

**Kuvaus:** Metodi jota kutsutaan kun XmlInsertissä on klikattu elementin, entiteetin tai id:n nimeä. Mikäli elementtilistassa klikattiin hiiren vasenta näppäintä, avataan dialogi johon voi syöttää attribuuttien arvoja ja mikäli elementtilistassa klikattiin hiiren oikeaa näppäintä, asetetaan pakolliset attribuutit tyhjiksi. Mikäli puunäkymässä on sisällönmuokkausikkuna aktiivisena, tehdään lisäys siihen kohtaan missä kursori sijaitsee siinä ja mikäli puunäkymässä on puu aktiivisena, tehdään lisäys tekstialueeseen.

**Parametrit:** MouseEvent evt - hiiritapahtuma

**Palautusarvo:** -

## Lisätyt sisäluokat:

### **NotValidElementDecl**

`NotValidElementDecl`

**Kuvaus:** Muodostin, joka asettaa parametrina tulleen elementtiniimen `elementName`-attribuutin arvoon.

**Parametrit:** `String aElementName` - elementin nimi

**Palautusarvo:** -

### **KeyHandler**

`keyPressed`

**Kuvaus:** Näppäimistönkäsittelijä, joka lisää `enter`-näppäimen painalluksella dokumenttiin listasta valitun elementin.

**Parametrit:** `KeyEvent evt`

**Palautusarvo:** -

## 5.2.5 **GeneralOptionPane.java**

### **Lisätyt attribuutit:**

- `JCheckBox showtextContent` - XML-pluginin options-välilehdellä oleva valintaruutu, joka antaa käyttäjän valita näytetäänkö elementtien tekstisisältö puunäkymässä vai ei.

`_init`

**Kuvaus:** Alustusmetodi, johon lisätty `showTextContent`in luominen.

**Parametrit:** -

**Palautusarvo:** -

`_save`

**Kuvaus:** Tallennusmetodi, johon lisätty `showTextContent`in arvon tallentaminen käyttäjän asetustiedostoon.

**Parametrit:** –

**Palautusarvo:** –

## 5.3 Muut muutokset

### 5.3.1 Parserin vaihto

Hakemistossa `jars` on tiedosto `xercesImpl.jar`, joka on Xetor-projektin vaihtama XML-parseri.

### 5.3.2 Lisätyt DTD:t

XML-pluginin `dtDs`-hakemistoon on lisätty `loose.dtd` ja `xhtml11.dtd`, jotta niitä ei tarvitsisi hakea Internetistä.

### 5.3.3 jEdit

`jEdit.java`-tiedoston `main`-metodiin on lisätty `initUserProperties`-metodin kutsu, `addPluginJAR`-metodiin `jar.uninit(true)`-metodikutsu ja `initSiteProperties`-metodiin valitun kielen alustus jotta saadaan suomenkieliset property-tiedostot toimimaan.

`PluginJAR.java`-tiedoston `generateCache`-metodiin on lisätty käyttäjän valitseman kielen tarkistus ja jos kieli on suomi, ladataankin `_fi.props`-päätteiset property-tiedostot.

### 5.3.4 Property-tiedostot

Lisäksi sovellukseen on lisätty seuraavat joko muokatut tai uudet property-tiedostot:

- `jedit_gui.props` (jEdit-hakemistossa)
- `jedit_gui_fi.props` `sidekick.props`
- `sidekick_fi.props`

### 5.3.5 Options

`GlobalOptions.java`-tiedostoon on lisätty `jEditGroup.addOptionPane("language")`-metodikutsu, joka lisää tiedostossa `LanguageOptionPane.java` olevan itse tehdyn `LanguageOptionPane`-luokan mukaisen välilehden `jEditin options`-valikkoon. `LanguageOptionPane` on peritty `jEditin AbstractOptionPanesta` ja siinä on toteutus alustusmetodille `_init` ja tallennusmetodille `_save`. Valittavina kielinä ovat Suomi ja Englanti. Näistä Englanti on oletuskieli ja käännös suomeksi on suurilta osin toteutettu.



## 6 Sovelluksen virheet

Tässä sovelluksessa listataan puutteet ja virheet sovellukseen toteutetuissa ominaisuuksissa.

Luku sisältää sovelluksen eri osissa havaittuja puutteita.

### 6.1 Puunäkymä

- Lisää joillekin elementeille ylimääräisiä attribuutteja
- Kontekstivalikon toiminnot ei toimi muilla kuin XML-dokumenteilla (ei testattu kunnolla)
- Puunäkymä ei pidä kaikkia avoimina olevia polkuja avoimina validoinnin jälkeen
- Puunäkymästä kopiointi ei käytä samaa leikepöytää kuin jEdit

### 6.2 Elementtilista

- Ehdottaa vain DTD:n mukaisia lisäyksiä, ei ota kantaa jo kirjoitettuun sisältöön
- Jos tekstinäkymässä siirretään kursoria ei elementtilista päivyty

## 7 Jatkokehitykseen siirretyt ominaisuudet

Tästä luvusta selviää ominaisuudet, joita projektin puitteissa ei ehditty toteuttaa.

### 7.1 CSS-editori

Alunperin oli tarkoituksena toteuttaa ominaisuuseditorin yhteyteen CSS-editori, joka näyttäisi valitun elementin CSS-ominaisuudet CSS-valitsimien mukaan ryhmiteltynä. Ajan puutteen vuoksi Xetor-projektin yhteydessä ei toteutettu kyseistä editoria.

Tämä tulee toteuttaa XML-plug-inin `XmlActions`-luokassa. Luokka toteuttaa tällä hetkellä ominaisuuseditorin attribuuttieditorin tehtävät.

### 7.2 Esikatselunäkymä

Esikatselunäkymä on toteutettu `InfoViewer`-plug-inissa, mutta ei näytä dokumentin ulkoasua standardin mukaisesti. Tähän pitäisi asentaa parempi selainkomponentti niin esikatselunäkymän saisi ankkuroitua jEditin käyttöliittymään.

### 7.3 Etsi/korvaa-toiminto

Sovellussuunnitelmassa [14] kuvattu etsi/korvaa-toiminto päätettiin jättää toteuttamatta. Etsi/korvaa-toiminnossa tulisi olla mahdollisuus valita korvataanko elementtien nimiä, attribuuttien nimiä, attribuuttien arvoja ja elementtien sisältöä.

### 7.4 jEditin kääntäminen suomen kielelle

Xetor-projekti aloitti jEditin käyttöliittymän kääntämisen suomenkieliseksi. Kaikkia käyttöliittymän toimintoja ei kuitenkaan ehditty suomentaa ja jo suoritettujen suomennosten oikeellisuutta ei ehditty tarkistaa. jEditin suomennokset löytyvät tiedostosta `jedit_gui.fi.props`, XML-plug-inin suomennokset tiedostosta `XML.fi.props` ja Sidekick-plug-inin suomennokset tiedostosta `Sidekick.fi.props`.

## 7.5 Muita pienempiä ominaisuuksia

- DTD:n generoiminen hyvinmuodostetusta dokumentista
- Puunäkymän kirjasinta ei pysty muuttamaan
- Sovellus sisältää valmiita dokumenttipohjia
- Puunäkymään ei voi määrittää pikanäppäimiä elementtien lisäykselle puuhun
- Puunäkymää ei voi tulostaa
- Puunäkymän väriä ei voi vaihtaa

## 8 Yhteenveto

Tässä dokumentissa kuvataan Xetor-projektiryhmän keväällä 2004 toteuttaman XML-editorin ominaisuuksia, rakennetta ja toimintaa. Toteutetuista ominaisuuksista kerrotaessa eritellään ne ominaisuudet, jotka ryhmä itse on toteuttanut, jEditin ja valmiiden plug-inien jo valmiiksi toteuttamista ominaisuuksista. Rakenteesta kertovassa luvussa luodaan yleiskuva jEditin käyttöliittymän toimintoihin. Xetor-plug-inien toiminnot kuvataan yksityiskohtaisesti, samoin kuin Xetor-ryhmän toteuttamien luokkien toteutus. Lisäksi dokumentissa on kuvattu sovellukseen jääneistä ja siinä havaituista virheistä sekä kerrottu jatkokehitykseen siirretyistä ominaisuuksista.

## Lähteet

- [1] World Wide Web Consortium, "Cascading Style Sheets, level 2 CSS2 Specification", <http://www.w3.org/TR/CSS2/>, 8-3-2004.
- [2] World Wide Web Consortium, "Document Object Model (DOM) Level 2 Core Specification", <http://www.w3.org/TR/2000/REC-DOM-Level-2-Core-20001113/>, 8-3-2004.
- [3] World Wide Web Consortium, "HyperText Markup Language (HTML) Home Page", <http://www.w3.org/MarkUp/>, 9-3-2004.
- [4] Sun Microsystems, "Java Technology", <http://java.sun.com/>, 9-3-2004.
- [5] Jaakohuhta Hannu, "Suuri tietotekniikan tietosanakirja", Suomen Atk-kustannus Oy, Helsinki, 1999.
- [6] Jaakohuhta, Hannu, "IT Ensyklopedia", IT Press, Helsinki 2001.
- [7] World Wide Web Consortium, "XHTML 1.0 The Extensible HyperText Markup Language (Second Edition)", <http://www.w3.org/TR/xhtml1/>, 9-3-2004.
- [8] World Wide Web Consortium, "Extensible Markup Language (XML) 1.0", <http://www.w3.org/TR/1998/REC-xml-19980210/>, 8-3-2004.
- [9] Ruini Henri, "Englanti - suomi -sanasto, v. 0.7 (XML)", <http://www.cs.helsinki.fi/u/ruini/structure/xml/sanasto.html>, 9-3-2004.
- [10] World Wide Web Consortium, "Extensible Stylesheet Language (XSL) Version 1.0", <http://www.w3.org/TR/xsl/>, 8-3-2004.
- [11] Xetor-projekti, "Vaatimusmäärittely", <http://sovellusprojektit.it.jyu.fi/xetor/dokumentteja/Vaatimusmaarittely.pdf>, 19-4-2004.
- [12] jEdit, "Open Source programmer's text editor", <http://www.jedit.org>, 21-4-2004.
- [13] jEdit, "jEdit 4.1 User's Guide", <http://www.jedit.org/users-guide/index.html>, 23-5-2004.
- [14] Xetor-projekti, "Sovellussuunnitelma", <http://sovellusprojektit.it.jyu.fi/xetor/dokumentteja/Sovellussuunnitelma.pdf>, 23-5-2004.