

Suunnitteludokumentti

Kaapo - Kaavioiden piirto-ohjelma

Helsinki 28.7.2005

Ohjelmistotuotantoprojekti

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

Kurssi

581260 Ohjelmistotuotantoprojekti (6 ov)

Projektiryhmä

Ilari Heikkinen

Allan Holsti

Tero Kallioinen

Kristian Ovaska

Mikko Paltamaa

Hannu-Pekka Rajaniemi

Asiakas

Inkeri Verkamo

Johtoryhmä

Juha Taina

Sampo Yrjänäinen

Kotisivu

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

Versiohistoria

Versio	Päiväys	Tehdyt muutokset
1.0	28.7.2005	Jäädetyt versio
0.95	27.7.2005	Tarkastuksen korjaukset tehty
0.9	25.7.2005	Tarkastettava versio
0.5	24.7.2005	Lisätty kaikkea
0.1	21.6.2005	Dokumentti luotu

Sisältö

1 Johdanto	1
1.1 Terminologiaa	1
1.2 Dokumentin rakenne	1
2 Arkkitehtuurin kuvaus	2
2.1 Arkkitehtuurikaavio	2
2.2 Päätoimintojen kuvaukset	2
2.2.1 Kaavion luonti, avaaminen ja tallentaminen	2
2.2.2 Elementin lisääminen kaavioon	3
2.2.3 Elementtien valitseminen, liikkuttaminen ja poistaminen	3
2.2.4 Elementin koon sekä elementtien ja yhteyksien attribuuttien arvojen muuttaminen	4
2.2.5 Yhteyden lisääminen kaavioon	6
2.2.6 Kaavioiden välillä liikkuminen	6
2.3 Pääkomponenttien rajapinnat	7
3 Komponentit	12
3.1 Project	12
3.1.1 <i>Project</i>	13
3.1.2 <i>ProjectComponent</i>	13
3.1.3 <i>DiagramComponent</i>	13
3.1.4 <i>Diagram</i>	13
3.1.5 <i>DiagramWrapper</i>	14
3.1.6 <i>Element</i>	14
3.1.7 <i>Connection</i>	14
3.1.8 <i>DiagramComponentWrapper</i>	14
3.1.9 <i>ElementWrapper</i>	14
3.1.10 <i>ConnectionWrapper</i>	14
3.2 ProjectManager	15
3.2.1 <i>ProjectManager</i>	15
3.2.2 <i>Selection</i>	15
3.2.3 <i>Project</i>	15
3.2.4 <i>UndoManager</i>	16

3.3	Command	16
3.3.1	<i>UndoableEdit</i>	16
3.3.2	<i>AbstractUndoableEdit</i>	16
3.3.3	<i>AbstractUndoableDComponentEdit</i>	16
3.3.4	<i>CompoundEdit</i>	16
3.3.5	<i>UndoManager</i>	16
3.3.6	<i>AddDComponentEdit</i>	18
3.3.7	<i>RemoveDComponentEdit</i>	18
3.3.8	<i>MoveDComponentEdit</i>	18
3.3.9	<i>ModifyDComponentAttributeEdit</i>	18
3.3.10	<i>MoveConnectionPointEdit</i>	18
3.3.11	<i>ResizeElementEdit</i>	18
3.3.12	<i>RemoveDiagramEdit</i>	18
3.3.13	<i>Edit</i>	18
3.3.14	<i>AbstractEdit</i>	18
3.3.15	<i>SelectionEdit</i>	19
3.3.16	<i>ChangeVisibleEdit</i>	19
3.3.17	<i>NewDiagramEdit</i>	19
3.4	TypeManager	19
3.4.1	<i>TypeManager</i>	19
3.4.2	<i>diagrams</i>	20
3.4.3	<i>diagramComponents</i>	20
3.5	FileManager	20
3.5.1	<i>FileManager</i>	21
3.5.2	<i>ObjectSaver</i>	21
3.5.3	<i>ObjectLoader</i>	21
3.5.4	<i>ExportPlugin</i>	21
3.5.5	<i>ComponentNotSupportedException</i>	21
4	Käyttöliittymäkuvaus	22
4.1	Perustyökaluvalikot	22
4.2	Kaaviokohtaiset työkalut	23
4.3	Projektin elementti- ja kaaviopuu	23
4.4	Piirtoalusta	24

4.5	Attribuuttipaneeli	24
5	Testaussuunnitelma	25
5.1	Testausstrategia	25
5.2	Testauksen vaiheet	25
5.2.1	Yksikkötestaus	25
5.2.2	Integroititestausta	25
5.2.3	Järjestelmätestaus	25
5.3	Testaamatta jäävä toiminnallisuus	26
5.4	Testausympäristö	26

Liitteet

1 Laajennetut käyttötapaukset

1 Johdanto

Oops on Helsingin yliopiston tietojenkäsittelytieteen laitoksen ohjelmistotuotantoryhmä. Tämä on ryhmän toteuttaman generisen kaavioiden piirto-ohjelmiston, Kaapon, suunnitteludokumentti. Dokumentissa esitellään järjestelmän komponentit rajapintoihin, kuvataan käyttöliittymä sekä esitellään testaussuunnitelma. Projektiin liittyvä materiaali on saatavissa ryhmän kotisivulta osoitteesta

`http://www.cs.helsinki.fi/group/oops`

1.1 Terminologiaa

- **Kääre (wrapper)**
Kaavion ilmentymä kaaviopuussa tai elementin tai yhteyden ilmentymä kaaviossa.
- **GUI**
Graphical User Interface, graafinen käyttöliittymä. Tässä dokumentissa tarkoittaa käyttöliittymän tarjoavaa pääkomponenttia.
- **Pääkomponentti**
Ohjelmiston osakokonaisuus, jolla on selkeä tehtävänrajaus, mutta joka ei kuitenkaan ole täysin itsenäinen kokonaisuus (osajärjestelmä). Pääkomponentti voi jakautua komponentteihinsa tai suoraan luokkiinsa.

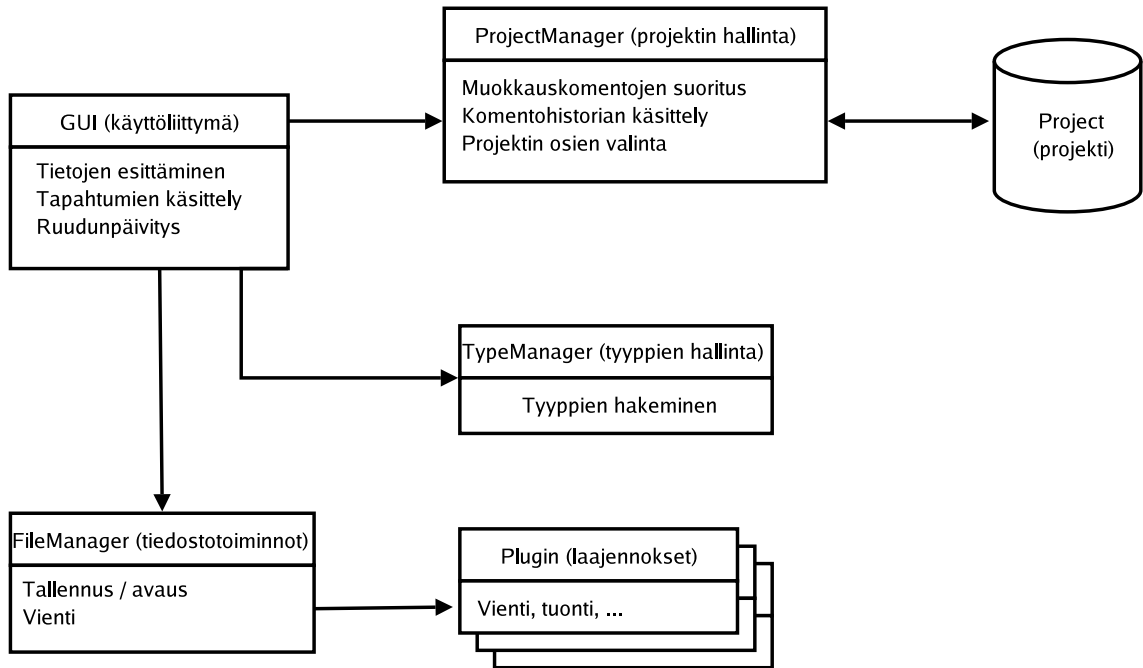
1.2 Dokumentin rakenne

Tässä luvussa, eli luvussa 1 on lyhyt kuvaus projektista, dokumentin termistö ja dokumentin rakenne. Luvussa 2 on esitelty ohjelmiston arkkitehtuuri arkkitehtuurikaavion ja pääkomponenttien rajapintojen muodossa. Lisäksi luvussa on kerrottu päätoimintojen kuvaukset sanallisessa ja sekvenssikaavioiden muodossa. Seuraavassa luvussa, eli luvussa 3 pääkomponentit on esitelty luokkakaavioiden ja -kuvausten tasolla. Käyttöliittymä on kuvattu luvussa 4, testaussuunnitelma luvussa 5 ja siihen liittyvät laajennetut käyttötapaukset luvussa liitteessä 1.

2 Arkkitehtuurin kuvaus

Tässä luvussa esitellään ohjelmiston arkkitehtuuri arkkitehtuurikaavion ja osajärjestelmien rajapintojen muodossa. Lisäksi luvussa on kuvattu päätoimintojen läpivienti sanallisesti ja sekvenssikaavioiden avulla.

2.1 Arkkitehtuurikaavio



Kuva 1: Arkkitehtuurikaavio

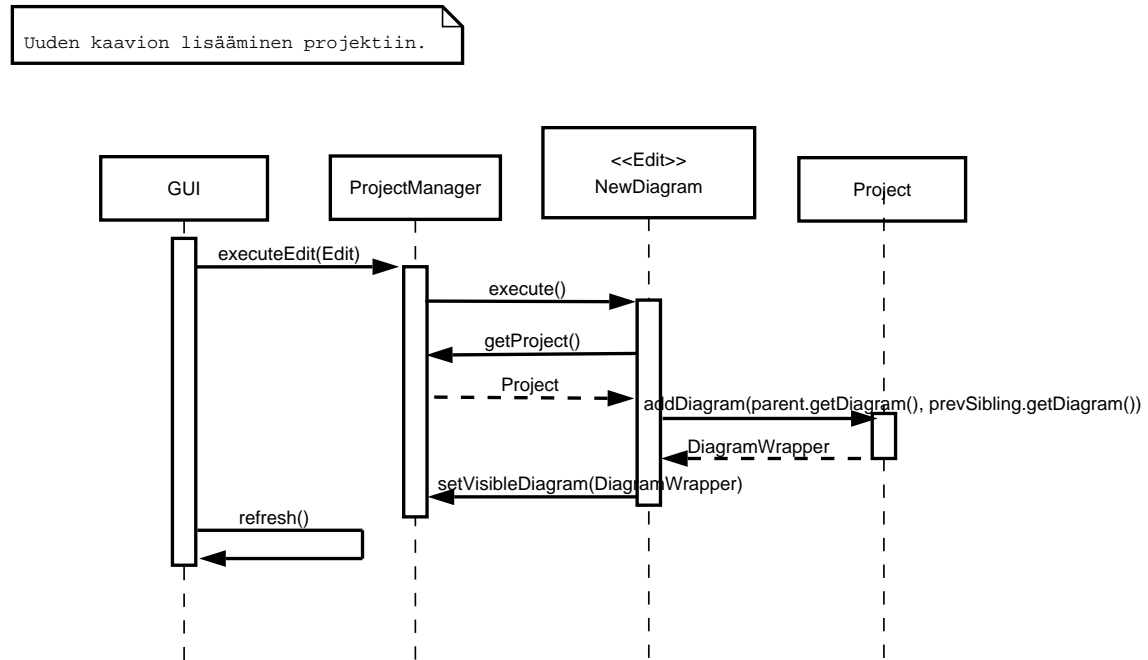
2.2 Päätoimintojen kuvaukset

Seuraavassa on kuvattu ohjelmiston päätoimintojen läpivienti arkkitehtuurin tasolla. Lisäksi liitteessä 1 on esitetty toimintoja tarkemmin kuvaavat sekvenssikaaviot.

2.2.1 Kaavion luonti, avaaminen ja tallentaminen

Uutta kaaviota luotaessa käyttöliittymä tarjoaa käyttäjän valittavaksi kaaviotyypit, jotka tyyppienhallinta on selvittänyt. Kun käyttäjä on valinnut kaaviotyypin, käyttöliittymä luo sopivan Edit-olion ja välittää sen projektinhallinnalle suoritettavaksi. Lopuksi käyttöliittymä päivittää työkalupalkkinsa vastaamaan kyseisen kaaviotyypin elementti- ja yhteysvalikoimaa. Kaaviot tallennetaan ja avataan projekteittain ja näistä toiminnoista huolehtii

tiedostonhallintakomponentti jolle käyttöliittymä välittää projektiolion ja tiedostonimen ja -polun.



Kuva 2: Kaavion lisääminen projektiin

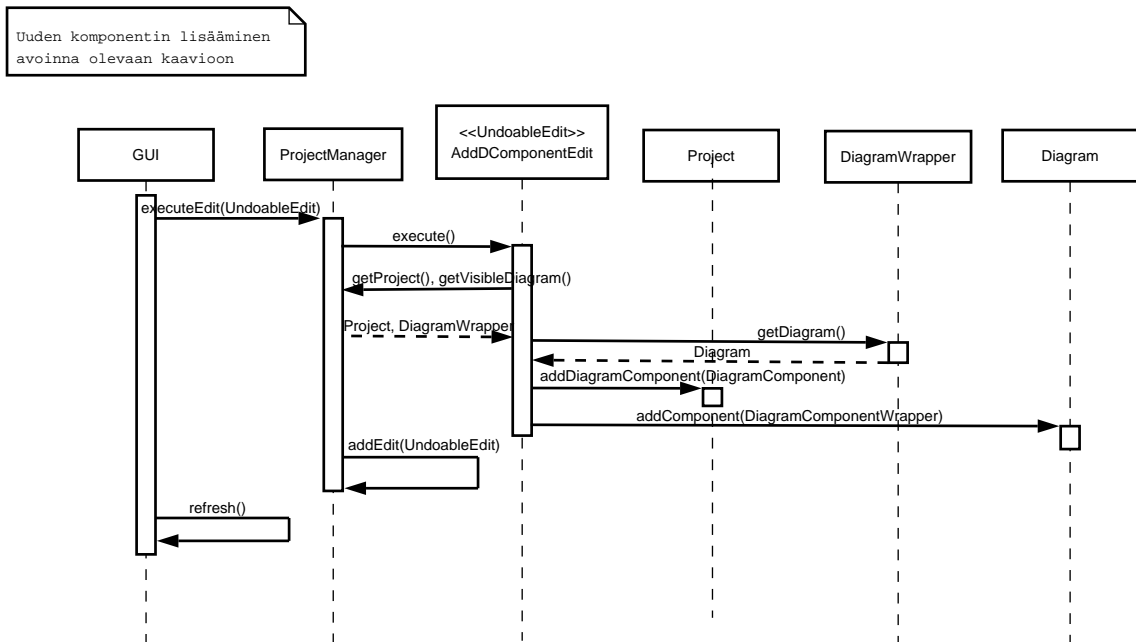
2.2.2 Elementin lisääminen kaavioon

Käyttöliittymä sisältää kyseisen projektin edustamaan kaaviotyyppeihin liittyvät elementtipainikkeet työkaluriviltään. Valittaessa jokin näistä painikkeista ja klikattaessa piirtoalustalle, käyttöliittymä luo lisäyksestä sopivan UndoableEdit-olion ja välittää sen projektinhallinnalle, joka taas suorittaa olion ja välittää sen mahdollista Undo-toimintoa varten UndoManagerille.

2.2.3 Elementtien valitseminen, liikuttaminen ja poistaminen

Kun piirtoalustalle klikataan tai vedetään hiirellä, käyttöliittymä projektinhallinnan avulla selvittää, oliko klikkaus- tai vetoalueella elementtejä. Jos klikkaus tai alueen valinta osui elementtiin, luodaan projektinhallinnassa Selection-olio ja tällöin kaikki muutokset, kuten liikuttamiset ja poistamiset kohdistuvat valittuihin elementteihin. Lisäksi käyttöliittymä piirtää valittujen elementtien ympärille valintaa korostavat kehykset. Jos valittuna on vain yksi elementti, käyttöliittymä kysyy elementiltä sille ominaista attribuuttipaneelia ja esittää sen.

Kun elementti on valittu, se voidaan vetää hiirellä eri kohtaan piirtoalustaa. Tässä käyttöliittymä luo sopivan UndoableEdit-olion ja välittää sen projektinhallinnalle. Projektin-



Kuva 3: Elementin lisääminen

hallinta suorittaa olion ja lisää sen UndoManagerin peruutusjonoon. UndoableEdit-olio itsessään tietää kuinka sen suoritus tapahtuu ja tekee tarvittavat kutsut projektin tietorakenteeseen.

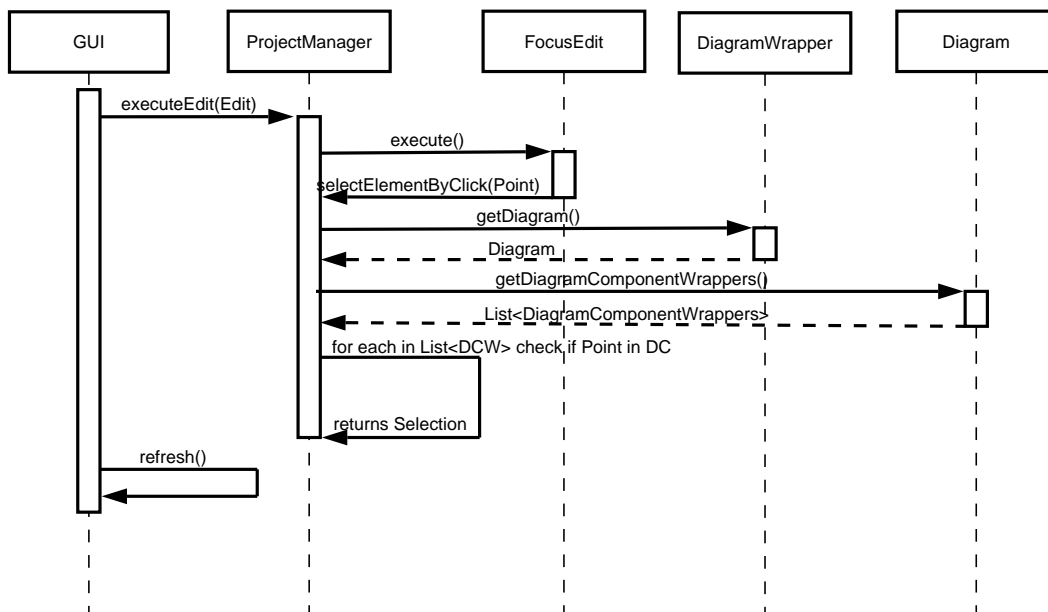
Valittu elementti voidaan myös poistaa. Tässä proseduuri on samankaltainen kuin liikuttamisessa.

2.2.4 Elementin koon sekä elementtien ja yhteyksien attribuuttien arvojen muuttaminen

Kun piirtoalustalla oleva elementti on valittu, käyttöliittymä piirtää sen ympärille "kahvat", joista hiirellä vetämällä elementin kokoa voi muuttaa. Tällöin käyttöliittymä luo sopivan UndoableEdit-olion, jonka projektinhallinta suorittaa ja lisää peruutettavien tapahtumien jonoon UndoManageriin.

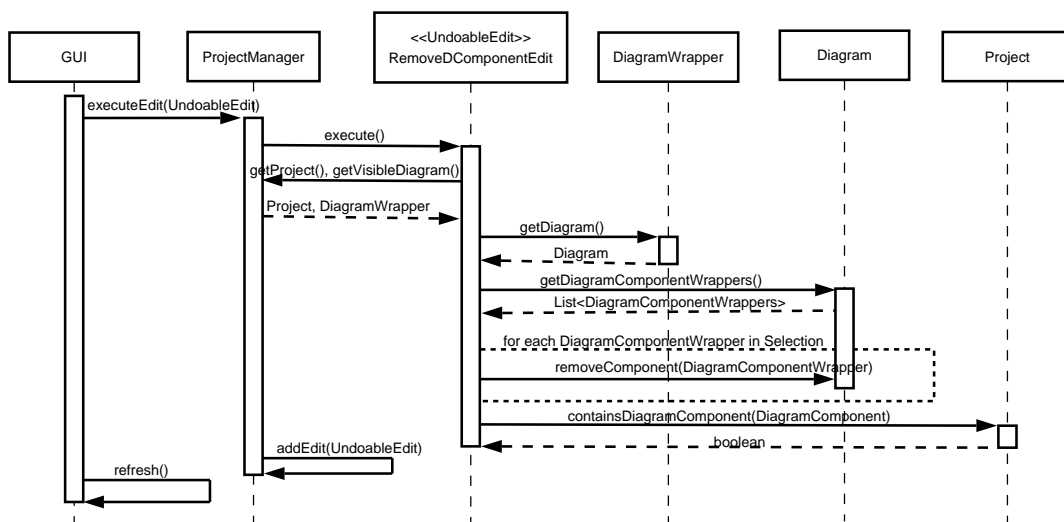
Kun piirtoalustalla oleva yksittäinen elementti tai yhteys on valittu, käyttöliittymä pyytää elementiltä siihen liittyvät attribuuttipaneelin. Attribuuttien arvojen muuttaminen tapahtuu paneelin tekstikenttiä muuttamalla ja kun muutos on valmis, projekti luo sopivan UndoableEdit-olion, joka suoritetaan projektinhallinnassa.

Elementin valinta



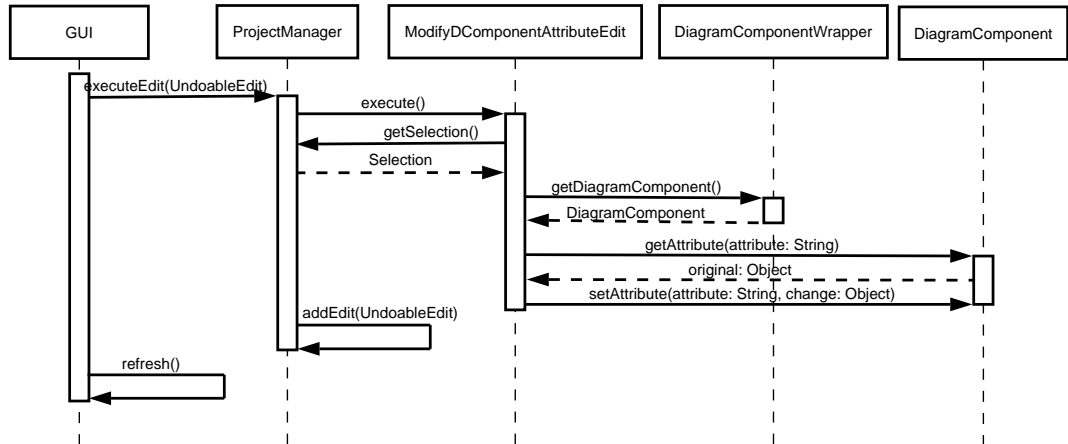
Kuva 4: Elementin valitseminen

Komponentin / komponenttijoukon poistaminen kaaviosta



Kuva 5: Elementin poistaminen

Kaavion elementin attribuutin muokkaaminen



Kuva 6: Attribuuttien muuttaminen

2.2.5 Yhteyden lisääminen kaavioon

Yhteyden lisääminen tapahtuu pääpiirteissään kuten elementinkin. Kun työkaluriviltä on valittu yhteys, käyttäjä valitsee klikkauksilla yhteyden jäsenet. Tällöin luodaan sopiva UndoableEdit-olio, joka suoritetaan projektinhallinnan toimesta.

2.2.6 Kaavioiden välillä liikkuminen

Eri projektien välillä liikkuminen tapahtuu valitsemalla käyttöliittymän kaaviopuusta haluttu kaavio. Valinnasta luodaan Edit-olio, joka suoritetaan projektinhallinnassa ja suorituksen yhteydessä projektinhallinta päivittää näkyvän kaavion kenttänsä. Lopuksi käyttöliittymä piirtää valitun kaavion.

2.3 Pääkomponenttien rajapinnat

Pääkomponentin nimi	FileManager (tiedostotoiminnot)
Kuvaus	Sisältää tiedostojen käsittelyyn liittyvät toiminnot, joita käyttöliittymä kutsuu.
Palvelut	
Palvelun nimi	saveProject
Palvelun kuvaus	Avaa parametrina annetun tiedoston ja kirjoittaa projektin tiedot siihen.
Sisäännotettavat parametrit	file : File, project : Project
Ulos annettavat tulokset	-
Poikkeustilanteet	IOException
Alkuehdot	-
Loppuehdot	Project-olio tallennettu tiedostoon
Palvelun nimi	openProject
Palvelun kuvaus	Avaa parametrina annetun tiedoston ja lukee projektin tiedot siitä.
Sisäännotettavat parametrit	file : File
Ulos annettavat tulokset	Project
Poikkeustilanteet	ClassNotFoundException
Alkuehdot	Tiedosto on olemassa ja oikean tyyppinen
Loppuehdot	-
Palvelun nimi	exportProject
Palvelun kuvaus	Vie projektin tiedostoon käyttäen parametrina annettua tiedostonimeä ja laajennosta.
Sisäännotettavat parametrit	file : File, plugin : ExportPlugin, project : Project
Ulos annettavat tulokset	-
Poikkeustilanteet	IOException
Alkuehdot	-
Loppuehdot	Project-olio viety tiedostoon
Palvelun nimi	getExportPlugins
Palvelun kuvaus	Palauttaa listan järjestelmään asennetuista viennitilaajennoksista.
Sisäännotettavat parametrit	-
Ulos annettavat tulokset	List<ExportPlugin>
Poikkeustilanteet	-
Alkuehdot	-
Loppuehdot	-

Pääkomponentin nimi	ExportPlugin (vientilaajennokset)
Kuvaus	Rajapinta vientilaajennoksille.
Palvelut	
Palvelun nimi	getName
Palvelun kuvaus	Palauttaa laajennoksen nimen.
Sisäännotettavat parametrit	-
Ulos annettavat tulokset	String
Poikkeustilanteet	-
Alkuehdot	-
Loppuehdot	-
Palvelun nimi	getFileExtension
Palvelun kuvaus	Palauttaa laajennoksen tiedostomuodon tiedostopäätteen.
Sisäännotettavat parametrit	-
Ulos annettavat tulokset	String
Poikkeustilanteet	-
Alkuehdot	-
Loppuehdot	-
Palvelun nimi	export
Palvelun kuvaus	Vie projektin soveltuvat tiedot tiedostoon jossain määrätyssä esitysmuodossa.
Sisäännotettavat parametrit	file : File, project : Project
Ulos annettavat tulokset	-
Poikkeustilanteet	IOException
Alkuehdot	-
Loppuehdot	-

Pääkomponentin nimi	ProjectManager (projektin hallinta)
Kuvaus	Sisältää projektin muokkaamiseen liittyvät palvelut, joita käyttöliittymä kutsuu.
Palvelut	
Palvelun nimi	getProject
Palvelun kuvaus	Palauttaa avoimena olevan projektin esim. tallennusta varten.
Sisäännotettavat parametrit	-
Ulos annettavat tulokset	project : Project
Poikkeustilanteet	-
Alkuehdot	-
Loppuehdot	-

Palvelun nimi	executeEdit
Palvelun kuvaus	Suorittaa parametrina annetun editointikomen- non projektille.
Sisäännotettavat parametrit	edit : Edit
Ulos annettavat tulokset	true jos käsky suoritettiin, muuten false
Poikkeustilanteet	-
Alkuehdot	-
Loppuehdot	Projektin tila on muuttunut
Palvelun nimi	executeEdit
Palvelun kuvaus	Suorittaa parametrina annetun editointikomen- non projektille, komento on myöhemmin ku- mottavissa.
Sisäännotettavat parametrit	edit : UndoableEdit
Ulos annettavat tulokset	true jos käsky suoritettiin, muuten false
Poikkeustilanteet	-
Alkuehdot	-
Loppuehdot	Projektin tila on muuttunut
Palvelun nimi	undo
Palvelun kuvaus	Palauttaa projektin historiassa edeltäneeseen ti- laan, mikäli sellainen on.
Sisäännotettavat parametrit	-
Ulos annettavat tulokset	true jos edeltävään tilaan voidaan palata, muu- ten false
Poikkeustilanteet	-
Alkuehdot	-
Loppuehdot	Edellinen toiminto kumottu
Palvelun nimi	canUndo
Palvelun kuvaus	Palauttaa tiedon siitä, voidaanko palata histo- riassa edeltäneeseen tilaan.
Sisäännotettavat parametrit	-
Ulos annettavat tulokset	true jos voidaan, muuten false
Poikkeustilanteet	-
Alkuehdot	-
Loppuehdot	-
Palvelun nimi	redo
Palvelun kuvaus	Palauttaa projektin peruutusta edeltävään tilaan, mikäli sellainen on.
Sisäännotettavat parametrit	-
Ulos annettavat tulokset	true jos edeltävään tilaan voidaan palata, muu- ten false
Poikkeustilanteet	-
Alkuehdot	-
Loppuehdot	Edellinen peruutus kumottu

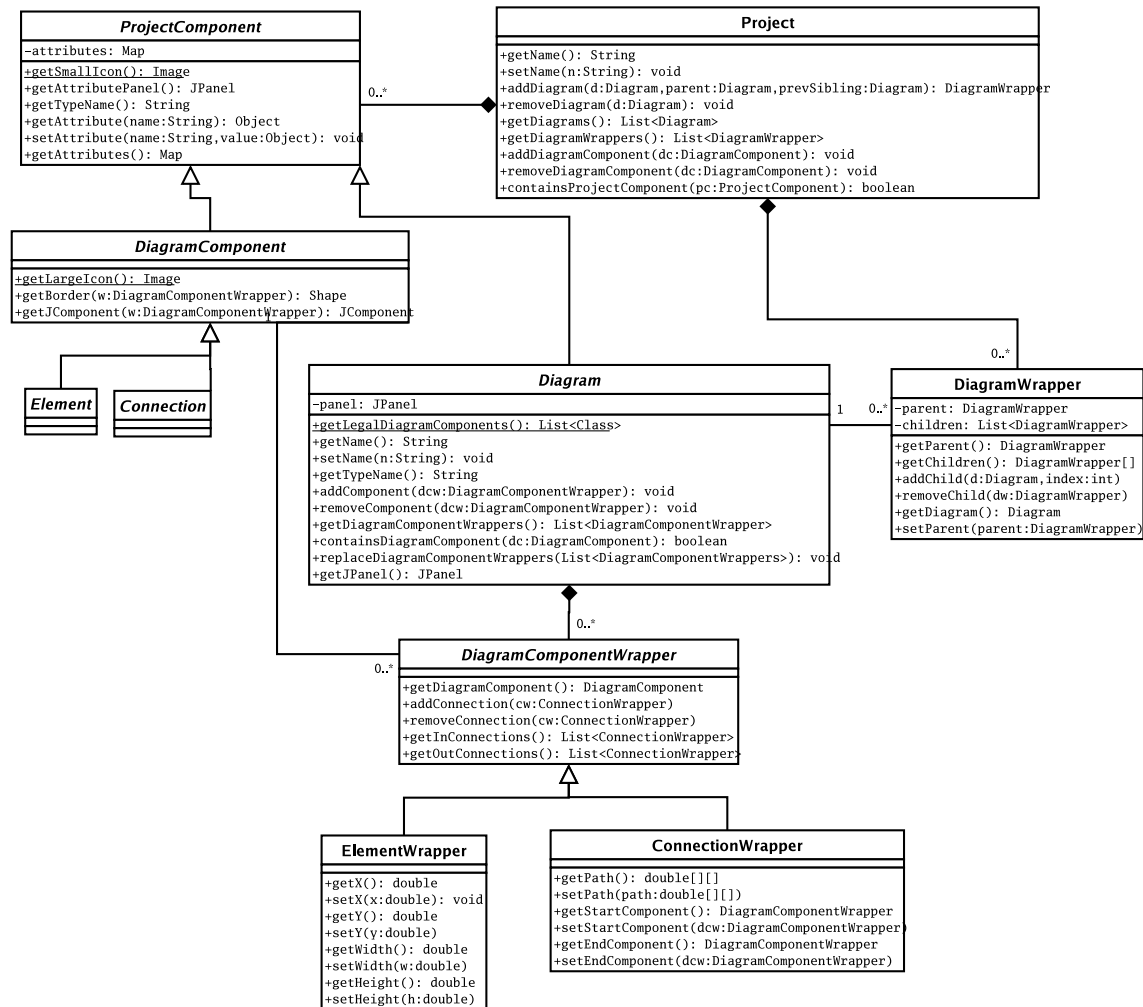
Palvelun nimi	canRedo
Palvelun kuvaus	Palauttaa tiedon siitä, voidaanko edetä historiassa seuraavaan tilaan.
Sisäännotettavat parametrit	-
Ulos annettavat tulokset	true jos voidaan, muuten false
Poikkeustilanteet	-
Alkuehdot	-
Loppuehdot	-
Palvelun nimi	setVisibleDiagram
Palvelun kuvaus	Asettaa kaavio-olion näkyvään tilaan
Sisäännotettavat parametrit	diagram : DiagramWrapper
Ulos annettavat tulokset	-
Poikkeustilanteet	DiagramException: kaavio ei kuulu projektiin
Alkuehdot	Projektissa on vähintään yksi kaavio
Loppuehdot	-
Palvelun nimi	setVisibleDiagram
Palvelun kuvaus	Palauttaa näkyvissä olevan kaavion
Sisäännotettavat parametrit	-
Ulos annettavat tulokset	DiagramWrapper
Poikkeustilanteet	-
Alkuehdot	Projektissa on vähintään yksi kaavio
Loppuehdot	-
Palvelun nimi	setSelection
Palvelun kuvaus	Ottaa muistiin viimeksi valittujen elementtien listan
Sisäännotettavat parametrit	selection : Selection
Ulos annettavat tulokset	-
Poikkeustilanteet	-
Alkuehdot	-
Loppuehdot	-
Palvelun nimi	getSelection
Palvelun kuvaus	Palauttaa valitut elementit sisältävän Selection-olion
Sisäännotettavat parametrit	-
Ulos annettavat tulokset	Selection
Poikkeustilanteet	-
Alkuehdot	-
Loppuehdot	-

Pääkomponentin nimi	TypeManager (tyyppien hallinta)
Kuvaus	Sisältää kaavio-, elementti- ja yhteystyyppien hallintaan liittyvät palvelut.
Palvelut	
Palvelun nimi	getDiagrams
Palvelun kuvaus	Palauttaa järjestelmässä käytettävissä olevat kaaviotyyppien luokat
Sisäänotettavat parametrit	-
Ulos annettavat tulokset	List<Class>
Poikkeustilanteet	-
Alkuehdot	-
Loppuehdot	-
Palvelun nimi	getDiagramComponents
Palvelun kuvaus	Palauttaa järjestelmän käytettävissä olevat elementti- ja yhteysluokat
Sisäänotettavat parametrit	-
Ulos annettavat tulokset	List<Class>
Poikkeustilanteet	-
Alkuehdot	-
Loppuehdot	-

3 Komponentit

Luvussa esitellään ohjelman pääkomponentit. Kustakin komponentista on kuvaus, luokkakaavio ja lyhyet luokkien kuvaukset.

3.1 Project



Kuva 7: Project-luokkakaavio

Project on tietorakennekomponentti, joka sisältää projektin kaaviot, elementit ja yhteydet. Pakkauksen luokat ProjectComponent, DiagramComponent ja Diagram sisältävät myös käyttöliittymätason toimintoja.

Loogisella tasolla projekti koostuu joukosta kaavioita (luokka Diagram), joukosta elementtejä ja yhteyksiä (luokka DiagramComponent ja sen aliluokat Element ja Connection) sekä kaaviopuusta (luokka DiagramWrapper).

Kaaviopuu on kaavioiden muodostama puumainen hierarkia, jossa ensimmäisellä tasolla on projektin päätason kaaviot ja kunkin kaavion alikaaviot ovat vanhempansa lapsisolmuja. Projekti-tietorakenteen kaaviopuu poikkeaa hieman käyttöliittymän kaaviopuusta (kts. luku 4): käyttöliittymä piirtää kaaviopuuhun myös kaavioiden elementit, mutta projekti-tietorakenteen kaaviopuussa on solmuina vain kaavioita. Kaavion elementteihin luonnollisesti päästään käsiksi kaavio-olion kautta.

Sekä kaaviot, elementit että yhteydet voivat esiintyä projektissa useaan kertaan, tämän vuoksi käytetään kääreluokkia (wrapper). Kaavio voi esiintyä projektissa useaan kertaan, jos se kuuluu kaaviopuuhun moneen kertaan. Elementti tai yhteys voi kuulua useaan kaavioon ja myös moneen kertaan samaan kaavioon.

Projektiin kuuluu siis kutakin kaaviota, elementtiä ja yhteyttä kohti yksi Diagram- tai DiagramComponent-luokan olio ja yksi tai useampi DiagramWrapper- tai DiagramComponentWrapper-luokan olio.

Project-komponentti huolehtii itse tietorakenteen lisäksi kaavioiden, elementtien ja yhteyksien graafisesta esityksestä. Kaavio esitetään piirtoalustalla JPanel-luokan oliona (metodi Diagram.getJPanel) ja elementit sekä yhteydet esitetään JComponent-luokan oliona (metodi DiagramComponent.getJComponent). Lisäksi kaaviot, elementit ja yhteydet palauttavat attribuuttipaneelin JPanel-oliona.

Kaikki luokat toteuttavat Serializable-rajapinnan.

3.1.1 *Project*

Project on koosteluokka, joka sisältää projektin kaaviot ja elementit. Project-olioon voi lisätä kaavion tiettyyn kohtaan kaaviopuussa ja poistaa kaavioita sekä pyytää kaaviopuu-tietorakenteen.

3.1.2 *ProjectComponent*

Kaavioiden, yhteyksien ja elementtien yhteinen ylliluokka. Luokan instansseilla on attribuuttikartta, joka sisältää avain-arvo pareja. Attribuutin avaimet ovat merkkijonoja ja arvot voivat olla mielivaltaisia olioita.

3.1.3 *DiagramComponent*

Elementtien ja yhteyksien yhteinen ylliluokka. DiagramComponent-olio osaa palauttaa käyttöliittymäkomponentin (getJComponent) sekä kertoa kaaviokomponentin ulkoreunat (getBorder), joita voi käyttää esim. yhteyspisteinä.

3.1.4 *Diagram*

Kaavioiden ylliluokka, josta kaikki konkreettiset kaaviotyypit perivät. Kaavio koostuu joukosta elementtien ja yhteyksien kääreitä (luokka DiagramComponentWrapper). Niitä voi

lisätä ja poistaa kaaviosta. Kaavioluokka osaa myös kertoa, mitä elementtejä siihen voi syntaksisääntöjen puitteissa lisätä. Luokan metodi `addComponent()` ei kuitenkaan estä vääränlaistenkaan elementtien lisäämistä.

3.1.5 *DiagramWrapper*

Luokka kuvaa kaavion tietyn esiintymän eli solmun kaaviopuussa. `DiagramWrapper`-oliot ovat puun solmuja, jotka tietävät vanhempansa ja lapsensa. Sama kaavio voi kuulua kaaviopuuhun useaan kertaan, jolloin puussa on useita vastaavia `DiagramWrapper`-luokan olioita.

3.1.6 *Element*

Elementtien abstrakti ylliluokka. Konkreettiset elementtityypit perivät tämän luokan.

3.1.7 *Connection*

Yhteyksien abstrakti ylliluokka. Konkreettiset yhteystyypit perivät tämän luokan.

3.1.8 *DiagramComponentWrapper*

Elementtien ja yhteyksien kääreluokkien ylliluokka. Kääre kuvaa elementin tai yhteyden tietyn esiintymän jossakin kaaviossa. Elementti voi kuulua useaan kaavioon (tai monta kertaa samaan kaavioon), jolloin sitä vastaavia kääreolioita on useita. Kääre sisältää sellaiset elementtiin liittyvät tiedot, jotka ovat kaaviokohtaisia. Esimerkiksi koko, sijainti sekä yhteyksien alku- ja loppupisteet voivat vaihdella elementin eri ilmentymissä eri kaavioissa. Yhteiset arvot, kuten elementin attribuutit, ovat `Element`- tai `Connection`-luokan oliossa, joita on vain yksi projektia kohden.

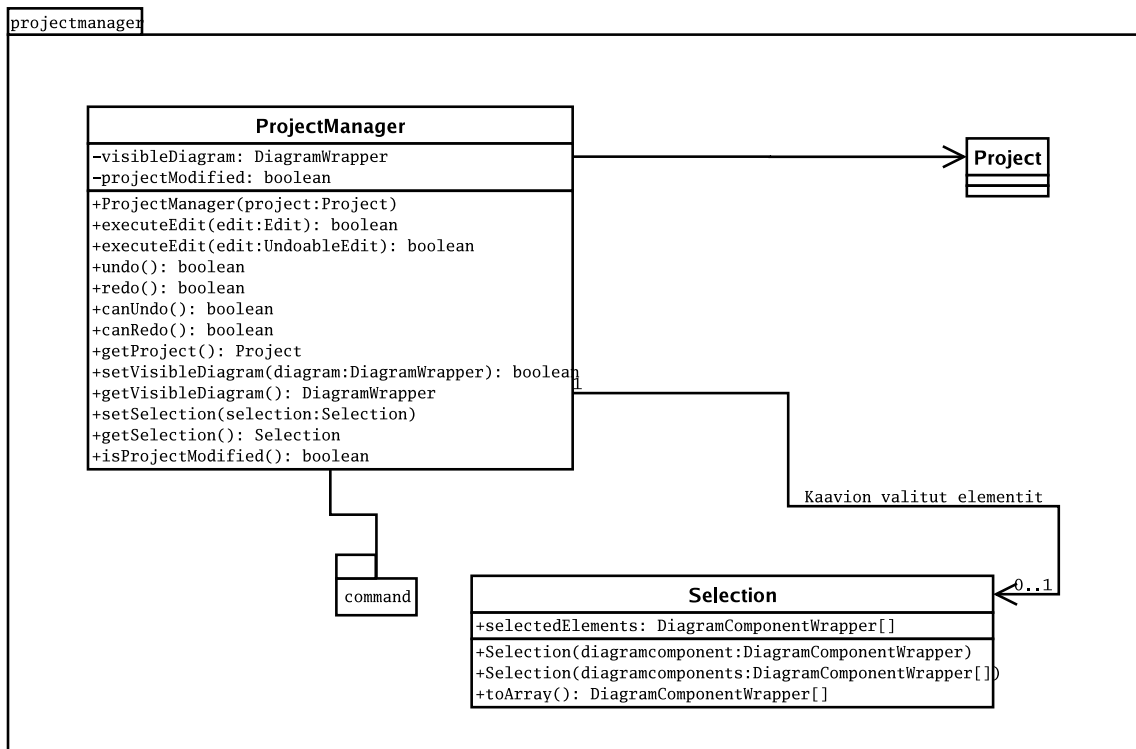
3.1.9 *ElementWrapper*

Konkreettinen luokka elementtien kääreille.

3.1.10 *ConnectionWrapper*

Konkreettinen luokka yhteyksien kääreille. Metodi `getPath()` palauttaa taulukon x- ja y-koordinaatteja, jotka kuvaavat yhteysviivan kulun kaaviossa.

3.2 ProjectManager



Kuva 8: ProjectManager -luokkakaavio

ProjectManager eli projektinhallinta on komponentti, joka toimii käyttöliittymän ja projektin tietorakenteen välissä hallintajärjestelmänä. Sen tehtäviin kuuluu käyttöliittymässä luotujen editointilioiden suorittaminen ja komentojen tarvitsemien tietorakenneviitauksen tarjoaminen. ProjectManager myös ylläpitää tilatietoa editoinnin kohteena olevasta kaaviosta ja editoinnin kohteena olevista elementeistä (Selection).

3.2.1 ProjectManager

Komponentin pääluokka, joka rajapintansa kautta tarjoaa kaikki komponentin palvelut.

3.2.2 Selection

Luokka Selection kuvaa editoitavan kaavion valittuna olevia elementtejä.

3.2.3 Project

Viittaus projektin tietorakenteeseen. Katso komponentti Project.

3.2.4 *UndoManager*

Viittaus UndoManager-olioon, joka liittyy peruutustoiminnon mahdollistavaan Command-osakomponenttiin. Kts. Command-komponentti.

3.3 Command

Command-alikomponentin ideana on saada aikaiseksi yksinkertainen rajapinta käyttöliittymän ja ProjectManagerin väliin sekä mahdollistaa undo- ja redo-toimintojen toteutus. Kun käyttäjä on määrittänyt jonkin kaaviota muokkaavan toiminnon GUI muodostaa siitä sopivan UndoableEdit-rajapinnan toteuttavan muokkausolion, jonka se antaa projectmanagerille. ProjectManager käynnistää toiminnon suorittamisen ja tämän jälkeen tallentaa toiminnon (mikäli tämä on Undoable) sisältämäänsä UndoManager-luokkaan.

Lisäksi komennoille, jotka eivät itsessään muokkaa kaaviota ja eivät täten ole peruutettavissa tai uudelleensuoritettaessa on oma Edit-luokkansa.

3.3.1 *UndoableEdit*

Javan rajapintaluokka `getPresentationName`-metodit. Rajapintaa käytetään piiloittamaan yksittäisten komentojen tarkempi tyyppi ProjectManagerissa.

3.3.2 *AbstractUndoableEdit*

Luokka toteuttaa UndoableEdit-rajapinnan määrittelemät metodit sekä Serializable-rajapinnan. Jokaisen toteutetun peruutettavissa olevan toiminnon yläluokka.

3.3.3 *AbstractUndoableDComponentEdit*

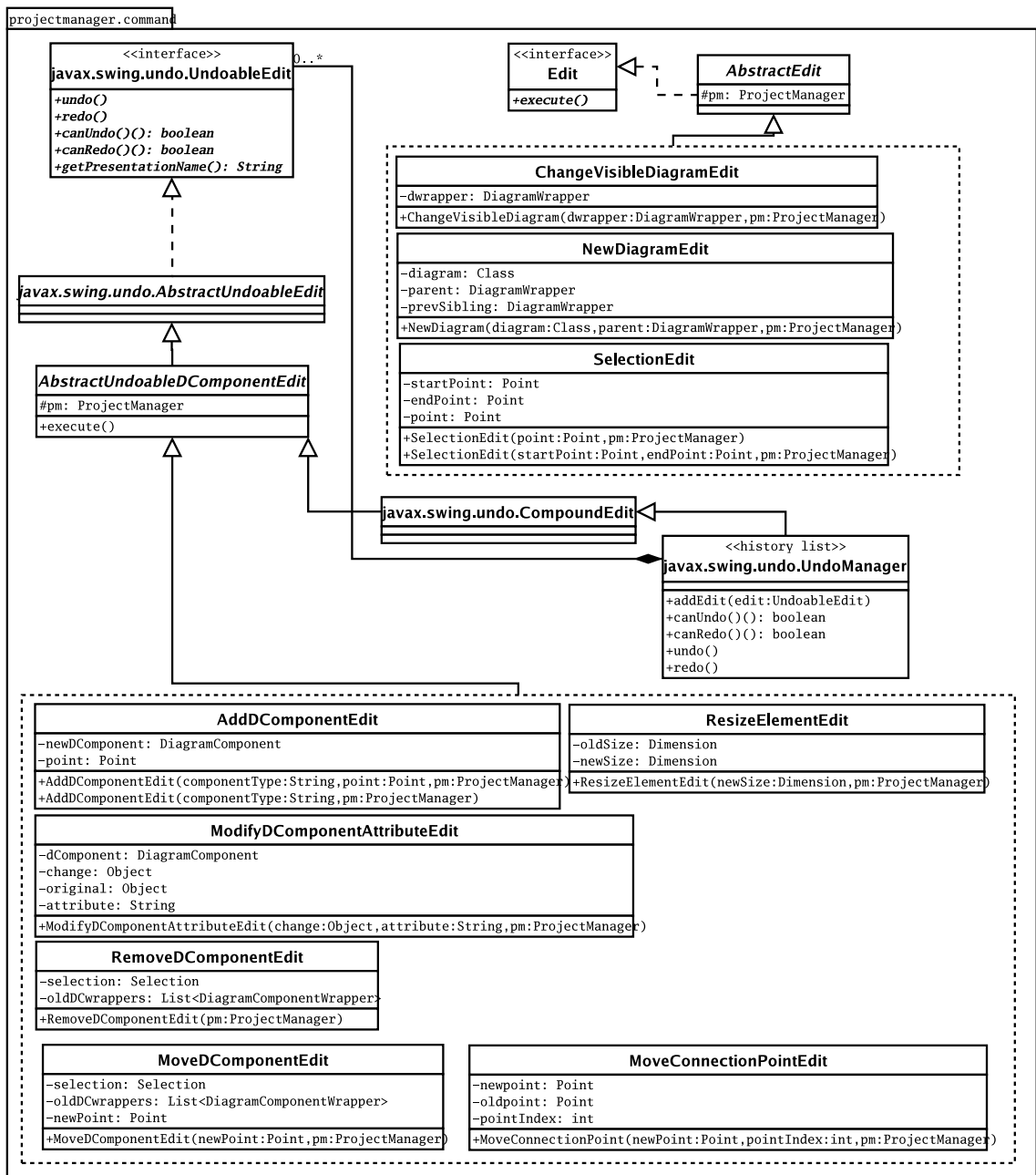
Luokka lisää jokaisessa UndoableEditissä tarvittavat omat metodit ja muuttujat.

3.3.4 *CompoundEdit*

Luokkaa käytetään kasaamaan useista eri UndoableEditeistä suurempia kokonaisuuksia.

3.3.5 *UndoManager*

UndoableEdit-luokan erikoistapaus, johon sisällytetään suoritettut komennot. Luokka hallinnoi sisällään undo- ja redo-jonoissa sille annettuja komentoja ja näiden vastaavia metodeja kutsutaan tämän luokan instanssin kautta.



Kuva 9: Command -luokkakaavio

3.3.6 *AddDComponentEdit*

Komento, joka suoritettaessa lisää hiirellä määriteltyyn paikkaan kaavioon uuden halutunlaisen komponentin.

3.3.7 *RemoveDComponentEdit*

Komento, jolla poistetaan kaaviosta halutut elementit. Komentoa peruutettaessa lisää poistettuja DiagramComponentWrapperit kaavion listaan.

3.3.8 *MoveDComponentEdit*

Komento, jolla liikutetaan kaavion elementtejä.

3.3.9 *ModifyDComponentAttributeEdit*

Komento, jolla muokataan jonkin komponentin attribuutteja.

3.3.10 *MoveConnectionPointEdit*

Komento, jolla liikutetaan yhteyden eri pisteitä päädyistä tai väliltä.

3.3.11 *ResizeElementEdit*

Komento, jolla muutetaan kaavioelementin kokoa. Ei käytetä yhteyksien muokkaamiseen.

3.3.12 *RemoveDiagramEdit*

Komento, jolla poistetaan projektista kaavio.

3.3.13 *Edit*

Peruuttamattomien komentojen rajapintaluokka. Vaatii sitä toteuttavilta luokilta komennon execute toteuttamista.

3.3.14 *AbstractEdit*

Peruuttamattomien komentojen abstrakti yläluokka, jossa määritellään jokaiseen aliluokkaan kuuluvaksi muuttujaksi viittaus ProjectManager-olioon.

3.3.15 *SelectionEdit*

Komento, jolla valitaan hiirellä määritellyltä alueelta fokus kaikkiin siihen kuuluviin elementteihin.

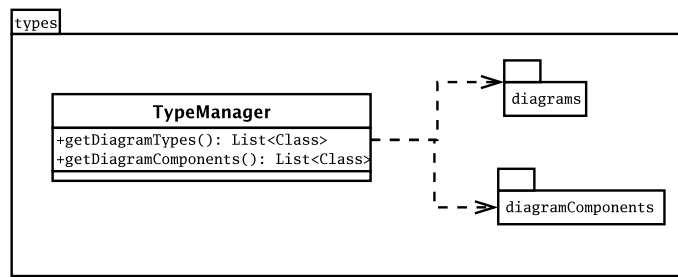
3.3.16 *ChangeVisibleEdit*

Komento, jolla vaihdetaan näkyvä kaavio toiseksi.

3.3.17 *NewDiagramEdit*

Komento, jolla luodaan uusi kaavio projektiin.

3.4 **TypeManager**



Kuva 10: TypeManager -luokkakaavio

TypeManager-luokkaa kutsutaan ohjelman lataamisen yhteydessä ennen kuin käyttöliittymä luodaan. Kuitenkin luokkaa kutsuu käyttöliittymä. Käyttöliittymä kutsuu luokan metodeja getDiagramTypes() ja getDiagramComponents(). getDiagramTypes()-metodi palauttaa käyttöliittymälle listauksen käytettävissä olevista kaaviotyypeistä, joista käyttöliittymä luo esityksen ruudulle ja samalla myös kykenee lataamaan luokan tarvittaessa käyttöönsä. getDiagramComponents()-metodilla käyttöliittymä pyytää käytettävissä olevat elementit ja yhteydet. Samastaan listasta käyttöliittymä voi tarvittaessa ladata käyttöönsä sopivat elementit ja yhteydet. TypeManager-luokka toteuttaa listan tuottamisen tutkimalla pakkausten sisältöä.

3.4.1 *TypeManager*

Hakee tiedostojärjestelmästä pakkauksiin asennetut kaavio- ja elementtityypit ohjelman käyttöön. Palauttaa kutsujalle listan kaavio- tai elementtiluokista. Komponenttia käyttää GUI. Luokka muodostaa listaukset tutkimalla pakkauksiin asennettuja luokkia. Se kerää

kaikki .class-päätteiset tiedostot ja muodostaa niistä Class-olioita, jotka se lisää linkitettyyn listaan. Listan toteutuksessa hyödynnetään Javan tarjoamaa geneerisyyttä varmistamaan jo käännoaikaisesti, etteivät listan objektit voi olla muuta muotoa kuin Class-olioita.

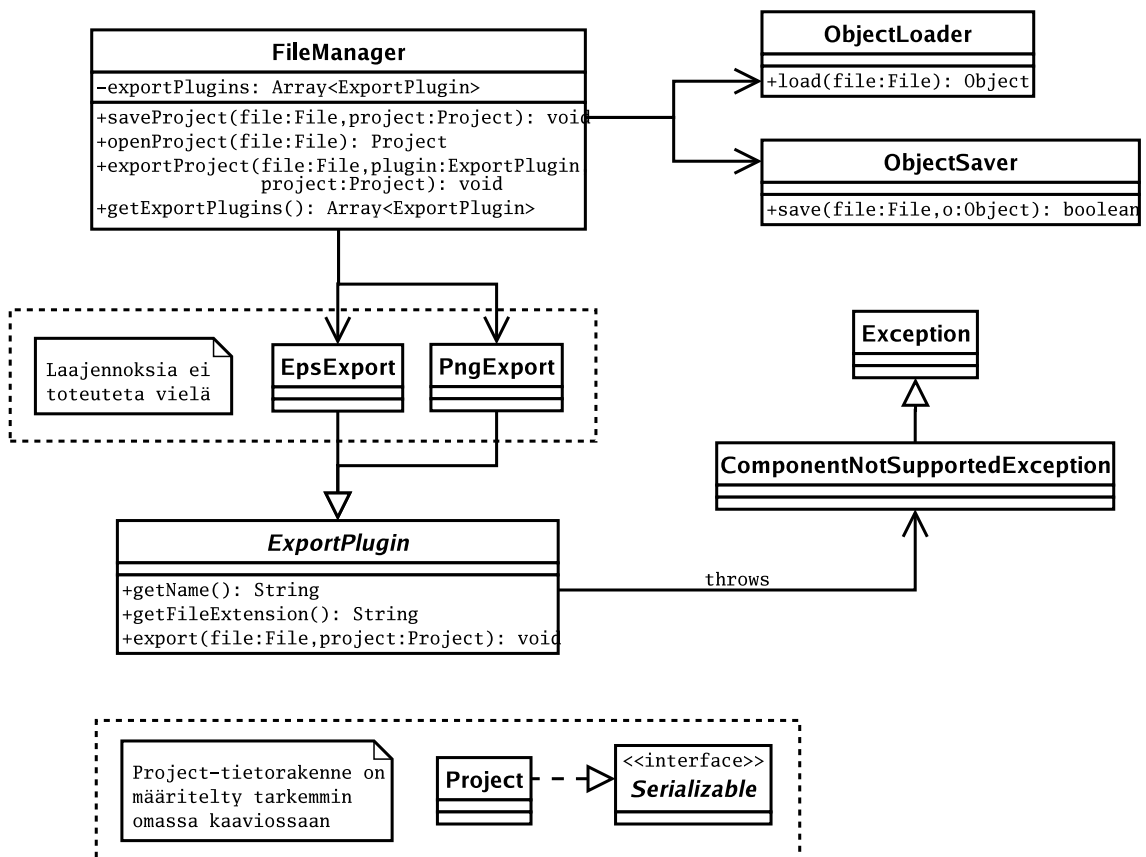
3.4.2 diagrams

Pakkaus sisältää ohjelmassa käytettävissä olevat kaaviotyypit.

3.4.3 diagramComponents

Pakkaus sisältää ohjelmassa käytettävissä olevat kaaviokomponentit eli yhteydet ja elementit.

3.5 FileManager



Kuva 11: FileManager -luokkakaavio

FileManager-luokkaa kutsuu käyttöliittymä, kun se haluaa tallentaa projektin tai avata tallennetun projektin. Käyttöliittymä kutsuu sitä myös saadakseen tiedon järjestelmään asennetuista vientilaajennoksista. Tallennettaessa projektia käyttöliittymä näyttää käyttäjälle tiedostovalintaikkunan, jolla käyttäjä määrittelee tiedoston nimen ja sijainnin. Dialogin Tallenna-painikkeen painamisen jälkeen kutsutaan FileManagerin metodia saveProject(), jolle annetaan parametrina tiedosto ja projekti. Avattaessa tallennettua projektia käyttöliittymä näyttää tiedostonvalintaikkunan, jolla käyttäjä voi valita avattavan tiedoston. Dialogin Avaa-painikkeen painamisen jälkeen kutsutaan FileManagerin metodia openProject(), jolle annetaan parametrina valittu tiedosto. Metodi palauttaa käyttöliittymälle uuden Project-olion. Vientiliitännäisten kanssa toimitaan samoin, eli käyttöliittymä huolehtii tiedoston valitsemisesta ja tarvittavien parametrien hankinnasta. Dialogin Vie-painikkeen painaminen johtaa exportProject()-metodin kutsumiseen valitulla tiedostonimellä ja viemiseen käytettävällä liitännäisellä. FileManager osaa myös antaa listan kaikista järjestelmään asennetuista vientiliitännäisistä.

3.5.1 FileManager

Avaa ja tallentaa Project-olion parametrina annetulla tiedostonimellä. Lisäksi luokka sisältää palvelut projektin vientiin käyttäen jotakin järjestelmään liitettyä vientiliitännäistä sekä järjestelmään liitettyjen vientiliitännäisten listaukseen. Käyttää tallentamiseen ja avaamiseen ObjectSaver- ja ObjectLoader-luokkia sekä järjestelmään asennettuja vientilaajennoksia.

3.5.2 ObjectSaver

Luokka Javan serialization-tallennustavan paketointiin. Sisältää metodin objektin tallentamiseen tallennusmedialle.

3.5.3 ObjectLoader

Luokka Javan serialization-tallennustavan paketointiin. Sisältää metodin objektin lataamiseen tallennusmedialta.

3.5.4 ExportPlugin

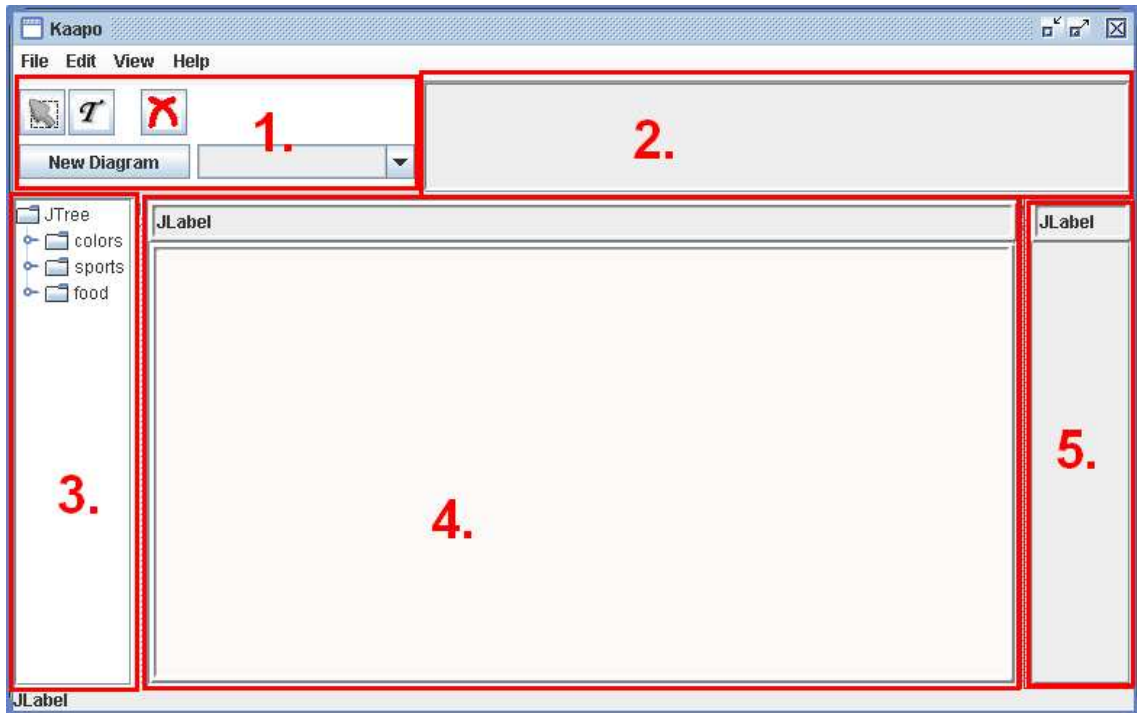
Abstrakti ylikuokka järjestelmään lisättäville vientiliitännäisille. Sisältää abstraktin metodin viemiseen sekä metodit tiedostopäätteen ja vientityypin nimen selvittämiseen.

3.5.5 ComponentNotSupportedException

Poikkeus, joka heitetään, jos liitännäinen ei pysty käsittelemään projektiin liittyvää oliota.

4 Käyttöliittymäkuvaus

Kuvassa 4 on kuvattu Kaapo-järjestelmän alustava käyttöliittymä ja sen eri osa-alueet. Kuvassa osa-alueet on rajattu ja numeroitu. Käyttöliittymähahmotelma perustuu vaatimusmäärittelyssä laadittuun käyttöliittymäprototyypin (kts. Vaatimusdokumentti).



Kuva 12: Käyttöliittymähahmotelma

4.1 Perustyökaluvalikot

Perustyökaluvalikko sisältää toiminnot, joilla on voidaan luoda projektiin uusia kaavioita. Valioissa on myös toiminnot valintojen tekemiseen, elementtien poistamiseen ja tekstin lisäämiseen kaavioon. Valikon toiminnot ovat käytettävissä kaikissa kaaviotyypeissä. (kts. kuva: alue 1.)

Select-näppäimen painaminen valitsee työkaluksi valinnan, eli hiiren kursori muuttuu kun se viedään piirtoalueelle. Valinta tapahtuma syntyy vasta kun hiiren painike nousee ylös piirtoalustalla ja valinta oli päällä. Toiminto toteutetaan PprojectManager-luokan metodeilla `selectElementByClick(point:Point)` tai `selectElementsByArea(startPoint:Point, endPoint:Point)`, jotka molemmat palauttavat Selection-olion.

Text-painikkeen painaminen valitsee teksti työkalun käyttöön, eli jos hiiri viedään nyt piirtoalueelle niin sen kursori muuttuu. Jos hiiri on piirtoalueella ja sillä klikataan niin se

luo kaavioon uuden tekstikentän.

Jos jotain on valittuna ja painetaan Delete-painiketta, valittu objekti poistetaan projektista. Valittu elementti saadaan ProjectManager-luokalta getSelection()-palvelulla. Poisto tapahtuu joko Project-luokan palvelulla removeDiagram(d:Diagram), jos poistettavana on kaavio tai Diagram-luokan metodilla removeComponent(dcw:DiagramComponentWrapper). Diagram-olion saa kysymällä Project-luokalta getVisibleDiagram().

Käyttöliittymän luonnin yhteydessä muodostetaan lista käytettävissä olevista kaaviotyypeistä pudotusvalikkoon. Lista saadaan kutsumalla TypeManager-luokan palvelua getDiagramTypes(). Listan saadaan nimet kun pyydetään saadusta class-olioita sisältävästä listasta kaikkien luokkien nimet.

New Diagram -painikkeen painaminen aiheuttaa tapahtuman, jossa projektiin luodaan uusi kaavio, joka on sitä tyyppiä, joka oli valittuna pudotusvalikosta. Toiminnon toteuttamiseksi käyttöliittymä pyytää ProjectManager-luokalta palvelua getProject(), joka palauttaa Project-olion. Tältä käyttöliittymä pyytää addDiagram(d:Diagram,parent:Diagram,prevSibling:Diagram) -palvelua. Parametrien toteuttamiseksi käyttöliittymä luo uuden Diagram-olion. Kun tiedot ovat selvillä se pyytää palvelun ja kaavio lisätään projektiin.

4.2 Kaaviokohtaiset työkalut

Työkalupakki sisältää avoimena olevan kaavion muokkaamiseen käytettävissä olevat työkalut, eli se sisältää mahdolliset yhteydet ja elementit. (kts. kuva: alue 2.)

Muodostaakseen kaaviokohtaisen työkalupalkin, käyttöliittymän tarvitsee pyytää ProjectManager-luokalta getVisibleDiagram()-palvelua. Tällä käyttöliittymä saa käyttöönsä näkyvän kaavion DiagramWrapper-olion, jolta se voi kysyä Diagram-olion ja siltä edelleen listan sallituista työkaluista. Tämän listan se saa käyttöönsä getLegalDiagramComponents()-palvelulla. Lista sisältää työkalut Class-olioina. Seuraavaksi käyttöliittymä lataa tarvittu luokat ja muodostaa nappulat. Nappuloihin se saa ikonit pyytämällä getLargeIcon()-palvelua, joka sijaitsee ProjectComponent-luokassa. Käyttöliittymä luo JButtonit ja lisää ne valikkoon.

4.3 Projektin elementti- ja kaaviopuu

Paneeli sisältää puuesityksen kaikista projektiin kuuluvista kaavioista ja niiden elementeistä. Puusta voidaan valita näkyvä kaavio samoin kuin siitä voidaan valita elementtejä. (kts. kuva: alue 3.)

Kun uusi projekti luodaan tai vanha avataan, paneeliin muodostetaan uusi puu projektiin kuuluvista kaavioista. Puuna käyteetään JTree-elementtiä. Käyttöliittymä hankkii tiedot esityksen luomiseksi pyytämällä ProjectManager-luokalta palvelua getProject(), joka palauttaa Project-olion käyttöliittymän käyttöön. Tältä käyttöliittymä pyytää projektiin kuuluvia kaavioita, elementtejä ja yhteyksiä. Ne se saa kutsumalla palvelua getDiagramWrappers(), joka palauttaa listan ylimmän tason DiagramWrapper-olioista. Näistä

voidaan muodostaa puun kaavioiden hierarkian ylin taso. Alikeaviot saadaan pyytämällä jokaiselle Wrapper-oliolle `getChildren()`. Näiden alikeaviot saadaan samoin. Elementtien hankkimiseksi on kutsuttava jokaisen `DiagramWrapper`-olion `getDiagram()`-palvelua, jolla saadaan `Diagram`-olio jolta pyydetään `getDiagramComponentWrappers()`-palvelua, joka palauttaa listan `DiagramComponentWrapper`-olioista. Palvelu peritään `Diagram`-luokalta. Näistä saadaan puuhun jokaisen kaavion komponentit. Kun kaavioon tehdään muutoksia, esim. lisätään uusi komponentti, ne päivitetään puuhun samaan tapaan kuin puun luonti mutta nyt riittää kun pyydetään valittuna olevan kaavion elementit. Samoin jos lisätään uusi kaavio niin riittää pyytää vain projektiin kuuluvien kaavioiden listaus.

4.4 Piirtoalusta

Piirtoalue muodostuu kaavion nimipalkista ja piirtotasosta. Nimipalkissa on editoitavan kaavion nimi. Piirtotasolla esitetään editoitava kaavio. Aluelle voidaan lisätä käytettäviä elementtejä ja yhteyksiä. Käytettävissä olevat työkalut saadaan työkalupalkista (kts. kuva: alue 2.). (kts. kuva: alue 4.)

Piirtoaluelle asetetaan `JPanel`, joka saadaan pyytämällä `ProjectManager`ilta `getVisibleDiagram()`, joka palauttaa `DiagramWrapper`-olion, jolta saadaan `Diagram`-olio. Tältä voidaan pyytää `getJPanel()`-metodilla tarvittava alusta. Alustan kuuntelu ja tarvittavien hiiren koordinaattien hankinta tapahtuu kuuntelemalla `JScrollPane`, johon `JPanel` asetetaan.

4.5 Attribuuttipaneeli

Paneeliin asetetaan joko kaavion tai valitun elementin tiedot. Tietoja voidaan editoida paneelia hyödyntäen. (kts. kuva: alue 5.)

Kun elementti tai yhteys valitaan, paneeliin vaihdetaan kyseisen elementin tiedot. Käyttöliittymä hankkii tiedot esityksen luomiseksi pyytämällä `ProjectManager`-luokalta palvelua `getSelection()`, joka palauttaa `DiagramComponent`-olion käyttöliittymän käyttöön. Koska se on `ProjectComponent`-luokan aliluokka niin siltä voidaan pyytää palvelua `getAttributePanel()`, joka palauttaa elementin tiedot `JPanel`ina. Se voidaan asettaa suoraan paikalleen.

Jos valintaa ei ole eli vain kaavio on valittuna niin tällöin näytetään kaavion tiedot, koska ne saadaan kun pyydetään `ProjectManager`-luokalta `getVisibleDiagram()`. Se palauttaa `DiagramWrapper`-olion, jolta pyydetään `Diagram`-olio, joka sekin on `ProjectComponent`-luokan aliluokka. Siltä saadaan palvelulla `getAttributePanel()` `JPanel`, joka sisältää kaavion tiedot ja se voidaan myös asettaa paikalleen suoraan.

5 Testaussuunnitelma

Suunnitelma sisältää kuvauksen testauksesta ja sen osavaiheista. Yksikkö- ja integrointi-testauksen yksittäiset testit suunnitellaan projektin toteutusvaiheessa. Järjestelmätestauksen käyttämät laajennetut testitapaukset löytyvät liitteestä 1.

5.1 Testausstrategia

Testaus jakaantuu yksikkötestaukseen, integrointitestaukseen ja järjestelmätestaukseen. Yksikkötestauksessa varmistutaan että kukin komponentti toimii sisäisesti oikein. Integrointitestauksessa varmistutaan siitä että kukin komponentti tarjoaa muille oikeanlaisen rajapinnan ja käyttää muita rajapintoja sovitusti. Järjestelmätestauksessa suoritetaan ohjelmalla ennalta määritellyjä käyttötapauksia joilla varmennetaan että ohjelma vastaa sille asetettuja, toteutettavaksi valittuja, toiminnallisia vaatimuksia. Testauksesta kirjoitetaan raportti johon kirjataan testien suoritus ja onnistuminen. Järjestelmätestauksen lisäksi erillistä hyväksymistestausta ei suoriteta ajallisten rajoitusten sekä toteutettavan ohjelman keskeneräisyyden vuoksi.

5.2 Testauksen vaiheet

5.2.1 Yksikkötestaus

Yksikkötestauksessa käytetään apuna JUnit testausohjelmaa. Koodaajat työskentelevät pareina siten että he kirjoittavat testit toistensa koodille. Testien suunnittelu on jokaisen parin omalla vastuulla.

5.2.2 Integrointitestaus

Integrointitestauksessa ohjelman erikseen koodatut yksiköt liitetään toisiinsa. Aina uuden yksikön liittämisen jälkeen testataan sen toiminta jo kootun kanssa.

5.2.3 Järjestelmätestaus

Järjestelmätestaus perustuu laajennetuihin käyttötapauksiin (lkt). Testit saadaan kunkin lkt:n päätöstaulukosta. Jokainen taulukon rivi määrittelee yhden ajettavan testin syötteen ja tulosteineen. Pienen poikkeuksen muodostavat Undo- ja Redo- tapaukset joille ei ole määritelty päätöstaulukoita. Nämä toiminnot testataan muitten testejen yhteydessä. Jokaisen testin jälkeen jonka toiminta kuuluu olla peruttavissa suoritetaan Undo- ja Redo- komennot. Mikäli Undo peruu ja Redo palauttaa testin lopputilan, hyväksytään testi myös Undo- ja Redo- kommentojen osalta.

5.3 Testaamatta jäävä toiminnallisuus

Testaus kattaa vain tärkeimmät toteutettaviksi valitut toiminnot. Tämän lisäksi testaamatta jäävät mahdollisesti ajan sallimat vaihtoehtoiset tavat suorittaa toimintoja sekä matalan prioriteetin toiminnot jotka toteutetaan vain osittain tai joihin vain varaudutaan toteutuksessa.

5.4 Testausympäristö

Ohjelma testataan laitoksen Linux-ympäristössä.

Liite 1. Laajennetut käyttötapaukset

LKT1		
Nimi		Kaavion luonti
Kuvaus		Käyttäjä luo uuden kaavion projektiin.
Syötteet (arvoalue)		Kaaviotyyppi (kaikkien määriteltyjen kaaviotyyppien joukko)
Tulosteet		Kaaviotyypin mukainen uusi kaavio piirtoalustalle ja projektipuuhun
Toimintasekvenssi		<ol style="list-style-type: none"> 1. Käyttäjä painaa New Diagram -painiketta 2. Ohjelma lisää uuden kaavion projektipuuhun ja piirtoalustalle 3. Tarvittaessa käyttäjä valitsee oikean kaaviotyypin alavetovalikosta 4. Ohjelma muuttaa uuden kaavion valitun tyyppiseksi
	Numero	Aikaisempi kaaviotyyppi
	1	Luokkakaavio
	2	Luokkakaavio
		Valittu kaaviotyyppi
		Luokkakaavio
		Käyttötapauskaavio

LKT2				
Nimi	Elementin lisääminen			
Kuvaus	Käyttäjä lisää haluamansa tyyppisen elementin kaavioon.			
Syötteet (arvoalue)	<ul style="list-style-type: none"> • Elementtityyppi (kaaviotyyppissä määriteltyjen elementtityyppien joukko) • X-koordinaatti (0:sta piirtoalustan maksimumileveyteen) • Y-koordinaatti (0:sta piirtoalustan maksimumikorkeuteen) 			
Tulosteet	Kaavio, johon on lisätty elementti			
Toimintasekvenssi	<ol style="list-style-type: none"> 1. Käyttäjä valitsee elementtityypin elementtipaletista 2. Ohjelma näyttää elementtityypin valittuna 3. Käyttäjä painaa hiiren painikkeella piirtoalustaa 4. Ohjelma lisää elementin kaavioon 			
	<u>Numero</u>	<u>Elementtityyppi</u>	<u>Koordinaatit</u>	<u>Ohjelman toiminta</u>
	1	Luokka	Keskellä	OK
	2	Luokka	min, min	Elementti piirtyy kokonaisena vas. ylänurkkaan ja piirtoalue laajenee
	3	Luokka	min, max	Elementti piirtyy kokonaisena oik. ylänurkkaan ja piirtoalue laajenee
	4	Luokka	max, min	Elementti piirtyy kokonaisena vas. alanurkkaan ja piirtoalue laajenee
	5	Luokka	max, max	Elementti piirtyy kokonaisena oik. alanurkkaan ja piirtoalue laajenee
	6	Luokka	Olemassaolevan elementin päällä	Elementti piirtyy vanhan päälle

LKT3			
Nimi		Elementin poistaminen	
Kuvaus		Käyttäjä poistaa elementin kaaviosta.	
Syötteet (arvoalue)		Poistettava elementti (kaaviossa olevat elementit)	
Tulosteet		Kaavio, josta elementti on poistettu	
Toimintasekvenssi		<ol style="list-style-type: none"> 1. Käyttäjä valitsee elementin piirtoalustalta 2. Käyttäjä painaa poista-painiketta 3. Ohjelma poistaa elementin kaaviosta 	
Numero	Elementti	Lisätiedot	Ohjelman toiminta
1	Luokka	Elementtiin ei tule yhteyksiä	OK
2	Luokka	Elementtiin tulee yhteyksiä	Yhteydet jäävät roikkumaan
3	Luokka	Kaavion viimeinen elementti	OK
4	Luokka	Elementti on useassa kaaviossa	Poistetaan vain valitusta kaaviosta

LKT4						
Nimi	Elementin koon muuttaminen					
Kuvaus	Käyttäjä muuttaa halutun elementin kokoa.					
Syötteet (arvoalue)	<ul style="list-style-type: none"> • Muutettava elementti (kaaviossa olevat elementit) • Uusi leveys (elementtikohtaisesta minimikoosta maksimiin) • Uusi korkeus (elementtikohtaisesta minimikoosta maksimiin) 					
Tulosteet	Kaavio, jossa elementillä on uusi koko					
Toimintasekvenssi	<ol style="list-style-type: none"> 1. Käyttäjä valitsee elementin piirtoalustalta 2. Käyttäjä vetää hiirellä elementin koonmuuttamispisteistä 3. Ohjelma piirtää elementin uudessa koossa 					
	Numero	Elementti	Lähtökoko	Valittu koko	Muuta	Toiminta
	1	Luokka	100, 100	150, 150	-	OK
	2	Luokka	100, 100	0, 0	-	Elementti ei pienene alle minimikoon
	3	Luokka	100, 100	max+1, max+1	-	Elementti ei suurene yli maksimikoon
	4	Luokka	100, 100	150, 150	Elementti reunan lähellä	Piirtoalusta laajenee
	5	Luokka	100, 100	150, 150	Elementti toisen vieressä	Elementti laajenee toisen päälle

LKT5				
Nimi	Elementin liikuttaminen			
Kuvaus	Käyttäjä liikuttaa elementtiä piirtoalustalle.			
Syötteet (arvoalue)	<ul style="list-style-type: none"> • Liikutettava elementti (kaaviossa olevat elementit) • Uusi x-koordinaatti (0:sta kaavion maksimileveyteen) • Uusi y-koordinaatti (0:sta kaavion maksimikorkeuteen) 			
Tulosteet	Kaavio, jossa elementin paikkaa on muutettu.			
Toimintasekvenssi	<ol style="list-style-type: none"> 1. Käyttäjä painaa hiiren painikkeen alas elementin kohdalla 2. Käyttäjä liikuttaa hiirellä elementin uuteen kohtaan 3. Ohjelma piirtää elementin uuteen kohtaan 4. Käyttäjä päästää hiiren painikkeen ylös 			
	Numero	Elementti	Koordinaatit	Ohjelman toiminta
	1	Luokka	Keskellä	OK
	2	Luokka	min, min	Piirtoalue laajenee antaen tilaa elementille
	3	Luokka	min, max	Piirtoalue laajenee antaen tilaa elementille
	4	Luokka	max, min	Piirtoalue laajenee antaen tilaa elementille
	5	Luokka	max, max	Piirtoalue laajenee antaen tilaa elementille
	6	Luokka	Olemassaolevan elementin päällä	Elementti piirtyy vanhan päälle

LKT6																															
Nimi	Elementin attribuuttien arvojen muuttaminen																														
Kuvaus	Käyttäjä valitsee elementin ja muuttaa sen attribuutin arvoa elementtipaneelin avulla tai suoraan piirtoalustalla.																														
Syötteet (arvoalue)	<ul style="list-style-type: none"> • Muokattava elementti (kaaviossa olevat elementit) • Muokattava attribuutti (elementin olemassaolevat attribuutit) • Attribuutin uusi arvo (elementin attribuutin sallitut arvot) 																														
Tulosteet	Kaavio, jossa attribuuttia on muutettu																														
Toimintasekvenssi	<ol style="list-style-type: none"> 1. Käyttäjä valitsee elementin 2. Käyttäjä valitsee muokattavan attribuutin attribuuttipaneelistä 3. Käyttäjä syöttää attribuutin uuden arvon 4. Ohjelma näyttää päivitetyn elementin 																														
	<table border="1"> <thead> <tr> <th>Numero</th> <th>Attr. tyyppi</th> <th>Vanha arvo</th> <th>Uusi arvo</th> <th>Muuta</th> <th>Toiminta</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>String</td> <td>Abc</td> <td>Def</td> <td>–</td> <td>OK</td> </tr> <tr> <td>2</td> <td>String</td> <td>(tyhjä)</td> <td>Abc</td> <td>–</td> <td>OK</td> </tr> <tr> <td>3</td> <td>String</td> <td>Abc</td> <td>Def</td> <td>Elementti useassa kaaviossa</td> <td>Arvo päivittyy kaikkiin</td> </tr> <tr> <td>4</td> <td>Int</td> <td>100</td> <td>xyz</td> <td>–</td> <td>Ohjelma valittaa eikä päivitä arvoa</td> </tr> </tbody> </table>	Numero	Attr. tyyppi	Vanha arvo	Uusi arvo	Muuta	Toiminta	1	String	Abc	Def	–	OK	2	String	(tyhjä)	Abc	–	OK	3	String	Abc	Def	Elementti useassa kaaviossa	Arvo päivittyy kaikkiin	4	Int	100	xyz	–	Ohjelma valittaa eikä päivitä arvoa
Numero	Attr. tyyppi	Vanha arvo	Uusi arvo	Muuta	Toiminta																										
1	String	Abc	Def	–	OK																										
2	String	(tyhjä)	Abc	–	OK																										
3	String	Abc	Def	Elementti useassa kaaviossa	Arvo päivittyy kaikkiin																										
4	Int	100	xyz	–	Ohjelma valittaa eikä päivitä arvoa																										

LKT7				
Nimi				Yhteyden lisääminen
Kuvaus				Käyttäjä lisää yhteyden kaavioon.
Syötteet (arvoalue)				<ul style="list-style-type: none"> • Yhteyden tyyppi (kaaviotyypin yhteystyypit) • Alkukoordinaatit (kaavion koordinaatisto) • Loppukoordinaatit (kaavion koordinaatisto)
Tulosteet				Kaavio, johon on lisätty yhteys
Toimintasekvenssi				<ol style="list-style-type: none"> 1. Käyttäjä valitsee elementtipaletista yhteystyypin 2. Käyttäjä valitsee hiirellä alkupisteen piirtoalustalta 3. Käyttäjä valitsee hiirellä loppupisteen piirtoalustalta 4. Ohjelma piirtää yhteyden
	Numero	Alkupää	Loppupää	Muuta
	1	Sisältää elementin	Sisältää elementin	–
	2	Ei sisällä	Sisältää	–
	3	Sisältää	Ei sisällä	–
	4	Ei sisällä	Ei sisällä	–
	5	Sisältää	Sisältää	Luvaton yhteys
	6	Sisältää	Sisältää	Välissä kolmas elementti
				Toiminta
				Yhteys elementtien välille
				Toispuoleinen yhteys
				Toispuoleinen yhteys
				Roikkuva yhteys
				Ohjelma huomauttaa virheestä
				Yhteys piirretty normaalisti

LKT8			
Nimi		Yhteyden poistaminen	
Kuvaus		Käyttäjä poistaa kaaviosta olemassaolevan yhteyden.	
Syötteet (arvoalue)		Poistettava yhteys (kaaviossa olevat yhteydet)	
Tulosteet		Kaavio, josta yhteys on poistettu	
Toimintasekvenssi		<ol style="list-style-type: none"> 1. Käyttäjä valitsee yhteyden 2. Käyttäjä painaa poista-painiketta 3. Ohjelma poistaa yhteyden 	
	Numero	Alkupiste	Loppupiste
	1	Kiinni elementissä	Kiinni elementissä
	2	Ei kiinni	Kiinni
	3	Kiinni	Ei kiinni
	4	Ei kiinni	Ei kiinni
			Toiminta
			Vain yhteys poistuu
			Vain yhteys poistuu
			Vain yhteys poistuu
			Yhteys poistuu

LKT9					
Nimi		Yhteyden uudelleenasetointi			
Kuvaus		Käyttäjä valitsee piirtoalustalla olevan yhteyden ja vetää sen uuteen paikkaan.			
Syötteet (arvoalue)		<ul style="list-style-type: none"> • Muokattava yhteys (kaaviossa olevat yhteydet) • Uusi alku- tai loppukoordinaatti (kaavion koordinaatisto) 			
Tulosteet		Kaavio, jossa yhteyden toinen pää muokattu			
Toimintasekvenssi		<ol style="list-style-type: none"> 1. Käyttäjä valitsee yhteyden piirtoalustalta 2. Käyttäjä tarttuu hiirellä yhteyden uudelleenasetointipisteeseen 3. Käyttäjä vetää pisteen uuteen paikkaan 4. Ohjelma piirtää muokatun yhteyden 			
Numero	Siirrettävä	Vanha piste	Uusi piste	Toiminta	
1	Alku	Kiinni elem.	Sisältää elem.	Alkupää	kiinni elementtiin
2	Alku	Kiinni	Ei sisällä	Alkupää	roikkumaan
3	Alku	Ei kiinni	Sisältää	Alkupää	kiinni elementtiin
4	Alku	Ei kiinni	Ei sisällä	Alkupää	jää roikkumaan
5	Loppu	Kiinni elem.	Sisältää elem.	Loppupää	kiinni elementtiin
6	Loppu	Kiinni	Ei sisällä	Loppupää	roikkumaan
7	Loppu	Ei kiinni	Sisältää	Loppupää	kiinni elementtiin
8	Loppu	Ei kiinni	Ei sisällä	Loppupää	jää roikkumaan

LKT10																															
Nimi	Yhteyden attribuuttien arvojen muuttaminen																														
Kuvaus	Käyttäjä valitsee yhteyden ja muuttaa sen attribuutin arvoa elementtipaneelin avulla tai suoraan piirtoalustalla.																														
Syötteet (arvoalue)	<ul style="list-style-type: none"> • Muokattava yhteys (kaaviossa olevat yhteydet) • Muokattava attribuutti (yhteyden olemassaolevat attribuutit) • Attribuutin uusi arvo (yhteyden attribuutin sallitut arvot) 																														
Tulosteet	Kaavio, jossa attribuuttia on muutettu																														
Toimintasekvenssi	<ol style="list-style-type: none"> 1. Käyttäjä valitsee yhteyden 2. Käyttäjä valitsee muokattavan attribuutin attribuuttipaneelistä 3. Käyttäjä syöttää attribuutin uuden arvon 4. Ohjelma näyttää päivitetyn yhteyden 																														
	<table border="1"> <thead> <tr> <th>Numero</th> <th>Attr. tyyppi</th> <th>Vanha arvo</th> <th>Uusi arvo</th> <th>Muuta</th> <th>Toiminta</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>String</td> <td>Abc</td> <td>Def</td> <td>–</td> <td>OK</td> </tr> <tr> <td>2</td> <td>String</td> <td>(tyhjä)</td> <td>Abc</td> <td>–</td> <td>OK</td> </tr> <tr> <td>3</td> <td>String</td> <td>Abc</td> <td>Def</td> <td>Yhteys useassa kaaviossa</td> <td>Arvo päivittyy kaikkiin</td> </tr> <tr> <td>4</td> <td>Int</td> <td>100</td> <td>xyz</td> <td>–</td> <td>Ohjelma valittaa eikä päivitä arvoa</td> </tr> </tbody> </table>	Numero	Attr. tyyppi	Vanha arvo	Uusi arvo	Muuta	Toiminta	1	String	Abc	Def	–	OK	2	String	(tyhjä)	Abc	–	OK	3	String	Abc	Def	Yhteys useassa kaaviossa	Arvo päivittyy kaikkiin	4	Int	100	xyz	–	Ohjelma valittaa eikä päivitä arvoa
Numero	Attr. tyyppi	Vanha arvo	Uusi arvo	Muuta	Toiminta																										
1	String	Abc	Def	–	OK																										
2	String	(tyhjä)	Abc	–	OK																										
3	String	Abc	Def	Yhteys useassa kaaviossa	Arvo päivittyy kaikkiin																										
4	Int	100	xyz	–	Ohjelma valittaa eikä päivitä arvoa																										

LKT11		
Nimi		Murtoviivan kulkureitin muuttaminen
Kuvaus		Käyttäjä muuttaa murtoviivan kulkureittiä.
Syötteen (arvoalue)		<ul style="list-style-type: none"> • Muokattava yhteys (kaaviossa olevat yhteydet) • Murtoviivan muokkauspiste (yhteysviivan muokkauspisteet) • Muokkauspisteen uusi paikka (kaavion koordinaatit)
Tulosteet		Kaavio, jossa yhteysviivan kulkureittiä on muokattu
Toimintasekvenssi		<ol style="list-style-type: none"> 1. Käyttäjä valitsee murtoviivan 2. Ohjelma piirtää murtoviivan alku-, loppu- ja muokkauspisteet 3. Käyttäjä vetää jonkin muokkauspisteen uuteen kohtaan 4. Ohjelma piirtää päivitetyn yhteysviivan
	Numero	Muokkauspisteitä
	1	1
	2	1
	3	2
	4	2
		Muokkaussuunta
		Vaaka
		Pysty
		Vaaka
		Pysty

LKT12				
Nimi				
Tekstin lisääminen kaavioon				
Kuvaus				
Käyttäjä lisää kaavioon vapaamuotoista tekstiä.				
Syötteet (arvoalue)				
<ul style="list-style-type: none"> • Tekstielementin koordinaatit (kaavion koordinaatisto) • Teksti 				
Tulosteet				
Kaavio, johon on lisätty tekstielementti				
Toimintasekvenssi				
<ol style="list-style-type: none"> 1. Käyttäjä valitsee työkaluvalikosta tekstityökalun 2. Käyttäjä valitsee piirtoalustalta tekstin paikan 3. Ohjelma näyttää kaaviossa vilkkuvan kursorin 4. Käyttäjä kirjoittaa tekstin kaavioon 				
Numero	Kaaviotyyppi	Tekstin paikka	Teksti	Toiminta
1	Luokkakaavio	Keskellä	abc	OK
2	Käyttötapauskaavio	Keskellä	abc	OK
3	Tietovuokaavio	Keskellä	abc	OK
4	Luokkakaavio	Elementin päällä	abc	OK

LKT13				
Nimi			Ryhmän valitseminen	
Kuvaus			Käyttäjä valitsee piirtoalustalla olevista elementeistä haluamansa ja muodostaa näin ryhmän joka voidaan liikuttaa ja poistaa	
Syötteet (arvoalue)			<ul style="list-style-type: none"> • Valittavat elementit (kaavion elementit) 	
Tulosteet			Kaavio jossa elementit ovat näkyvästi valittu	
Toimintasekvenssi			<ol style="list-style-type: none"> 1. Käyttäjä painaa shift-näppäimen alas 2. Käyttäjä valitsee elementit yksitellen 3. Ohjelma korostaa käyttäjän valitsemat elementit 	
	Numero	Elementtien lkm	Tyyppi	Toiminta
	1	2	Möykkyjä	Elementit liikkuvat ja poistuvat kokonaisuutena
	2	1	Möykky	Elementti toimii yksittäisenä valittuna elementtinä
	3	10	Möykkyjä	Elementit liikkuvat ja poistuvat kokonaisuuten
	4	2	Yhteyksiä	Elementit poistuvat kokonaisuutena mutta eivät liiku
	5	1	Yhteys	Elementti toimii yksittäisenä valittuna elementtinä
	6	10	Yhteyksiä	Elementit poistuvat kokonaisuutena mutta eivät liiku
	7	10	Sekalaisia	Elementit poistuvat ja liikkuvat kokonaisuutena

LKT14			
Nimi		Alueen rajaaminen	
Kuvaus		Käyttäjä valitsee alueella olevat elementit rajaustyökalua käyttäen	
Syötteet (arvoalue)		<ul style="list-style-type: none"> • Alkupiste (kaavion koordinaatisto) • Loppupiste (kaavion koordinaatisto) 	
Tulosteet		Kaavio jossa alueen elementit ovat näkyvästi valittu	
Toimintasekvenssi		<ol style="list-style-type: none"> 1. Käyttäjä valitsee rajaustyökalun työkalupalkista 2. Käyttäjä valitsee hiirellä piirtoalustalta aloituspisteen 3. Käyttäjä valitsee hiirellä piirtoalustalta loppupisteen 4. Ohjelma korostaa määritellyn alueen rajaamat elementit valittuina 	
	Numero	Alue	Elementtejen Toiminta lkm
	1	Kekseltä reunalle	5 Alueen rajaamat elementit valittu
	2	Reunalta keskelle	5 Alueen rajaamat elementit valittu
	3	Keskellä iso	8 Alueen rajaamat elementit valittu
	4	Keskellä olematon	0 Ei toimintaa
	5	Koko kaavio	10 Kaikki kaavion elementit valittu

LKT15	
Nimi	Undo
Kuvaus	Käyttäjä haluaa perua suorittamansa toiminnon. Testataan kunkin käyttötapauksen testauksen yhteydessä joka voidaan perua. Jokaisen tällaisen laajennetun käyttötapauksen päätöstaulukosta valitaan yksi tyypillinen rivi jonka jälkeen suoritetaan undo-komento. Jos tämä palauttaa käyttötapauksen alkutilaan niin hyväksytään.
Syötteet (arvoalue)	
Tulosteet	
Toimintasekvenssi	

LKT16	
Nimi	Redo
Kuvaus	Käyttäjä haluaa perua suorittamansa perumisen. Testataan kunkin undo-testin yhteydessä. Suoritetaan redo ja mikäli käyttötapauksen lopputila palautuu niin hyväksytään.
Syötteet (arvoalue)	
Tulosteet	
Toimintasekvenssi	

LKT17										
Nimi	Tallennus vanhaan tiedostoon									
Kuvaus	Käyttäjä haluaa tallentaa projektin sille aikaisemmin määriteltyyn paikkaan									
Syötteet (arvoalue)	Ei syötteitä									
Tulosteet	Alariville tieto onnistuneesta tallennuksesta									
Toimintasekvenssi	<ol style="list-style-type: none"> 1. Käyttäjä valitsee File-valikosta Save-komennon 2. Ohjelma tallentaa projektin sen vanhaan tiedostoon 									
	<table border="1"> <thead> <tr> <th>Numero</th> <th>Tiedosto</th> <th>Toiminta</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Määritelty</td> <td>Projekti tallentuu tiedostoon</td> </tr> <tr> <td>2</td> <td>Ei määritelty</td> <td>Save-komento harmaannettu</td> </tr> </tbody> </table>	Numero	Tiedosto	Toiminta	1	Määritelty	Projekti tallentuu tiedostoon	2	Ei määritelty	Save-komento harmaannettu
Numero	Tiedosto	Toiminta								
1	Määritelty	Projekti tallentuu tiedostoon								
2	Ei määritelty	Save-komento harmaannettu								

LKT18		
Nimi	Tallennus uuteen tiedostoon	
Kuvaus	Käyttäjä haluaa tallentaa projektin uuteen tiedostoon	
Syötteet (arvoalue)	<ul style="list-style-type: none"> • Uuden tiedoston nimi • Uuden tiedoston sijainti 	
Tulosteet	Alariville tieto onnistuneesta tallennuksesta	
Toimintasekvenssi	<ol style="list-style-type: none"> 1. Käyttäjä valitsee File-valikosta Save As -komennon 2. Ohjelma avaa tiedostoikkunan 3. Käyttäjä valitsee polun ja uuden tiedoston nimen 4. Ohjelma luo tiedoston ja tallentaa projektin siihen 	
	Numero	Tiedosto
	1	Uusi
	2	Vanha
		Toiminta
		Projekti tallentuu tiedostoon
		Ohjelma kysyy varmennusta ennen tallentamista

LKT19			
Nimi		Tiedoston avaus	
Kuvaus		Käyttäjä haluaa avata tiedostoon tallennetun projektin	
Syötteet (arvoalue)		<ul style="list-style-type: none"> • Tiedoston nimi 	
Tulosteet		Projekti avautuu ohjelmassa	
Toimintasekvenssi		<ol style="list-style-type: none"> 1. Käyttäjä valitsee File-valikosta Open File -komennon 2. Ohjelma avaa tiedostoikkunan 3. Käyttäjä valitsee avattavan tiedoston ikkunasta 4. Ohjelma avaa projektin muokattavaksi 	
Numero	Tiedosto	Muuta	Toiminta
1	Oikea tyyppi	-	Tiedoston projekti avautuu ohjelmaan
2	Väärä tyyppi	-	Ohjelma ilmoittaa virheellisestä tyypistä
3	Virheellinen	-	Ohjelma ilmoittaa kelvottomoasta tiedostosta
4	Oikea tyyppi	Tallentamaton projekti avoinna	Ohjelma antaa käyttäjälle mahdollisuuden tallentaa avoin projekti ennen avaamista