

JHS 162 Paikkatietojen mallintaminen tiedonsiirtoa varten

Liite 1 UML-mallinnus

Versio: 2.0

Julkaistu: 31.10.2011

Voimassaoloaika: toistaiseksi

Sisällys

1 Yleistä.....	1
2 Lyhenteet.....	1
3 UML-luokkakaaviotekniikan perusteet.....	1
4 Ohjeita paikkatietojen mallintamiseen.....	6
5 Tiedonsiirron sovelluskeemaan liittyvät lisäohjeet.....	6

1 Yleistä

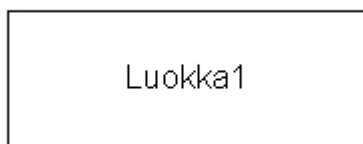
ISON paikkatietoja koskevassa standardoinnissa - sekä itse standardointityössä että myös konkreettisten sovelluskohtaisten tietomallien laatimisessa - käytetään Unified Modeling Language (UML, versio 2.1) -mallinnuskieleen kuuluvaa luokkakaaviotekniikkaa. Standardi *ISO/TS 19103 Conceptual schema language* ohjaa UML-kaaviotekniikan käyttämistä paikkatietomallinnuksessa. Lisäksi standardit *SFS-EN ISO 19109 Rules for application schema* ja *EN ISO 19136 Geography Markup Language* antavat lisäohjeita UML-mallien laatimiselle. Tässä dokumentissa annettavat yleiset ohjeet UML-luokkakaaviotekniikan soveltamisesta paikkatietojen mallintamiseen perustuvat UML:n perussääntöihin ja em. standardeihin.

2 Lyhenteet

CSL: Conceptual Schema Language
ISO: International Organization for Standardization
GML: Geography Markup Language
UML: Unified Modeling Language

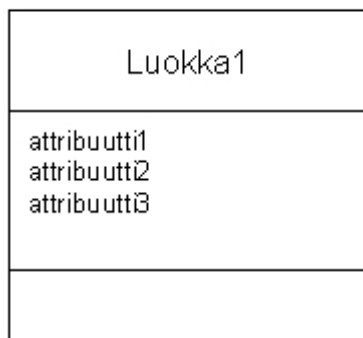
3 UML-luokkakaaviotekniikan perusteet

Luokka esitetään suorakaiteena.



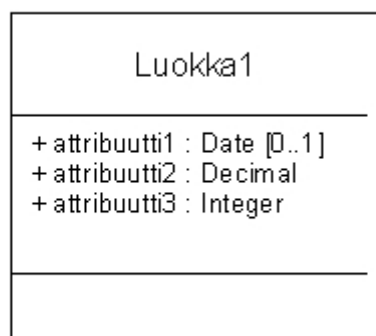
Luokan attribuutit (esim. attribuutti1) sijoitetaan suorakaiteen sisälle luokan nimestä väliviivalla erotettuna.

JUHTA - Julkisen hallinnon tietohallinnon neuvottelukunta

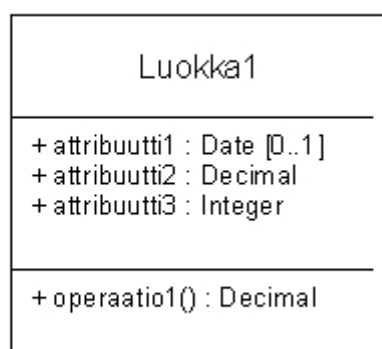


Attribuutti voidaan määritellä julkiseksi (+ -merkki attribuutin nimen edessä), jolloin sen näkyy luokan ulkopuolelle. Attribuutille voidaan antaa tietotyyppi (esim. Date) ja toistuvuusmääre. Toistuvuus voidaan ilmoittaa yhdellä luvulla tai arvovälinä esimerkiksi seuraavasti:

- [1] esiintyy vain kerran, [3] esiintyy kolme kertaa.
- [0..1] valinnainen – ei esiinny lainkaan tai esiintyy kerran.
- [1..*] tai [*] vapaasti toistuva – esiintyy yhden tai useamman kerran.



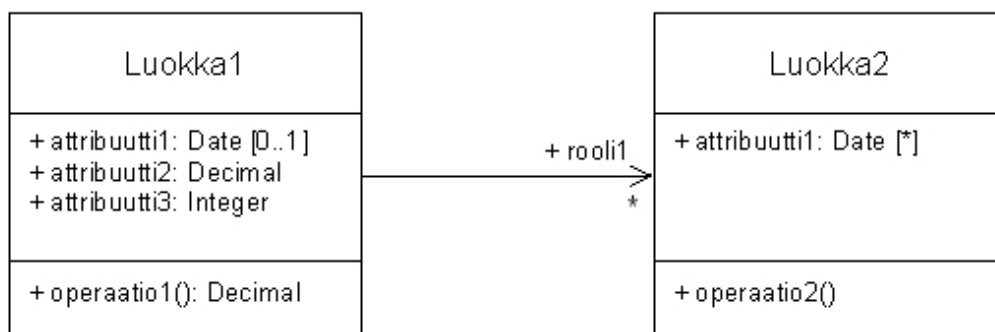
Luokalle voi myös määrittää operaatioita (esim. operaatio1). Nämä kirjoitetaan attribuuttien alapuolelle niistä väliviivalla erotettuna.



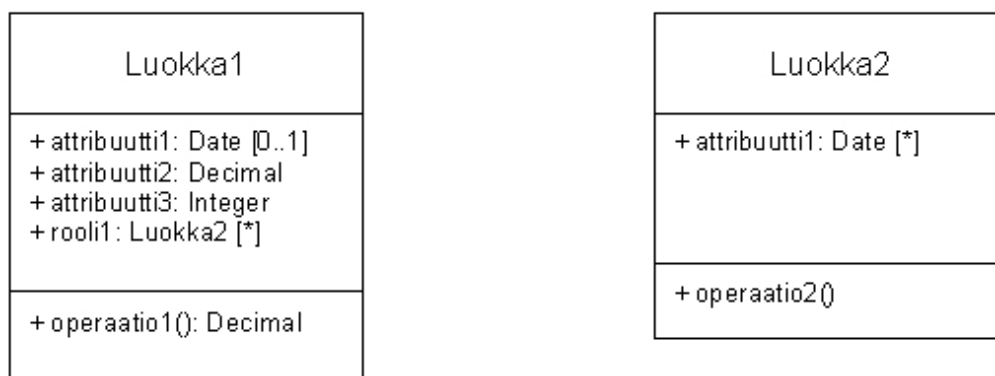
Luokkien välillä voi olla suhde. Tämä osoitetaan luokkien väliin piirretyllä viivalla. Suhde on joko yksisuuntainen (osoitettu viivan päässä olevalla nuolella) tai kaksisuuntainen (tavallisesti molemmat nuolen päät jätetään piirtämättä). Kohdeluokan (Luokka2) rooli lähdeluokalle (Luokka1) ilmaistaan viivan viereen

JUHTA - Julkisen hallinnon tietohallinnon neuvottelukunta

sijoitetulla roolinimellä (rooli1), joka sijoitetaan viivan kohdeluokan puoleiseen päähän. Suhteen toistuvuus osoitetaan toistuvuusmerkinnällä (esim. *).

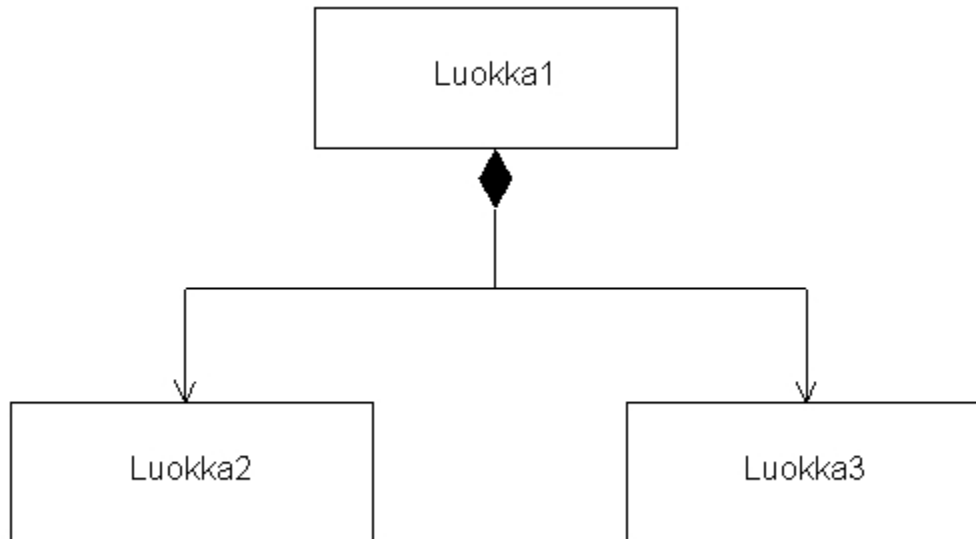


Luokkien välinen suhde voidaan osoittaa myös lähdeluokan (Luokka1) attribuutilla (rooli1), jonka tietotyyppinä on kohdeluokka (Luokka2). Seuraava malli on merkitykseltään identtinen edellisen mallin kanssa. Edellinen merkintätapa on havainnollisuutensa vuoksi suositeltavampi. Jälkimmäisestä tavasta selviää kuitenkin paremmin, että suhde on samanlainen luokan ominaisuus kuin attribuutti.



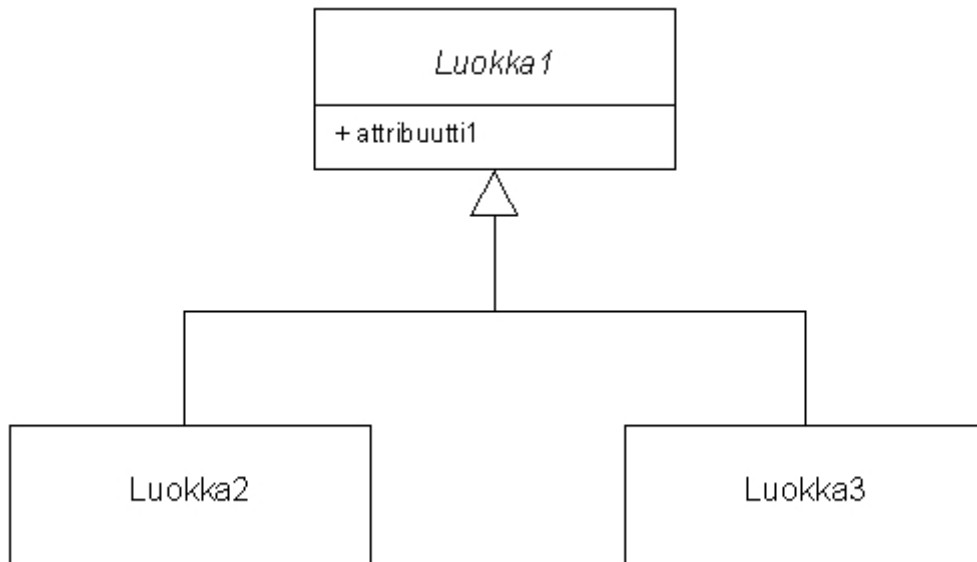
Koostesuhteessa luokka (Luokka1) voi koostua toisista luokista (Luokka2 ja Luokka3) ja tämä osoitetaan koosteella (komposiitti). Tällöin komponenttiluokan (Luokka2, Luokka3) ilmentymä voi olla vain koosteluokan (Luokka1) yhden ilmentymän osa. Komponenttiluokan ilmentymä ei voi esiintyä ilman koosteluokan ilmentymää.

JUHTA - Julkisen hallinnon tietohallinnon neuvottelukunta



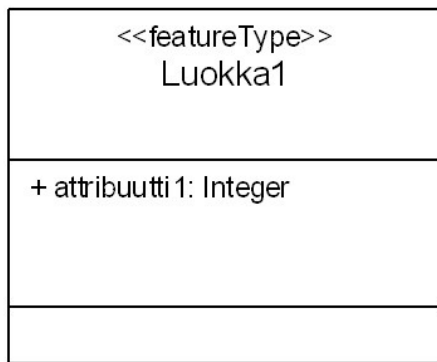
Luokkien välillä voi vallita perintähierarkia. Luokka (aliluokka) voi periä toisesta luokasta (yliluokka, kantaluokka). Perintähierarkia esitetään luokkien väliin piirrettävän viivan yläluokan puoleiseen päähän sijoitettavalla avoimella nuolenpäällä. Aliluokka (Luokka 2, Luokka3) perii kaikki yläluokalle (Luokka1) määritellyt ominaisuudet (attribuutti1). Datasollla aliluokan ilmentymä voi aina esiintyä yläluokan ilmentymän sijasta.

Perintähierarkian esittämisessä käytetään usein abstrakteja luokkia. Abstraktista luokasta ei ole ollenkaan ilmentymiä – kyseessä on siis pelkästään tietomallissa näkyvä käsite. Luokan abstraktius osoitetaan kirjoittamalla sen nimi vinokirjaimin (*Luokka1*).



Luokan (Luokka1) luonnetta voidaan osoittaa antamalla sille stereotyyppi (featureType), joka sijoitetaan luokan nimen yläpuolelle kaksoiskulmasulkujen sisään (esimerkiksi kuvassa oleva stereotyyppi 'featureType' osoittaa, että luokka edustaa mallissa reaali maailman kohdeluokkaa). Stereotyyppi kuvaa luokkien samankaltaisuutta ilman suhdetta tai perintää.

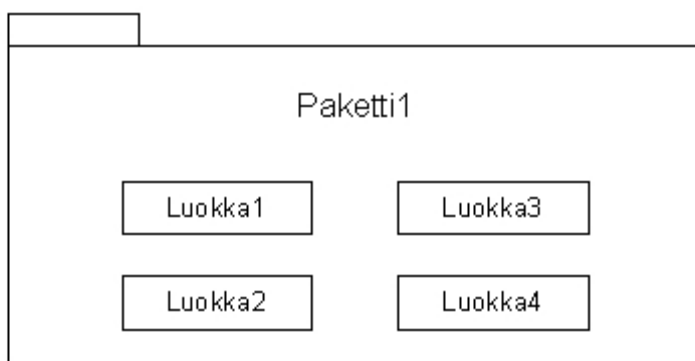
JUHTA - Julkisen hallinnon tietohallinnon neuvottelukunta



Luokat voidaan ryhmitellä paketeiksi. Pakettien välillä voi olla riippuvuuksia, jotka osoittavat, että tietyn paketin (Paketti1) luokat riippuvat tavalla tai toisella toisen paketin (Paketti2) luokista.



Joskus myös pakettiin kuuluvat luokat piirretään näkyviin.



4 Ohjeita paikkatietojen mallintamiseen

Paikkatietojen mallintamisessa tulee ottaa huomioon seuraavia seikkoja:

- Paikkatietokohteet mallinnetaan UML-luokkina.
- Paikkatietokohteiden välinen suhde mallinnetaan
 - UML-suhteena, jos suhteeseen ei liity omia ominaisuuksia.
 - Itsenäisenä luokkana (suhdeluokka), jos suhteeseen liittyy ominaisuuksia.
- Paikkatietokohteiden väliset suhteet, jotka vastaavat kokonaisuus/osa -suhdetta mallinnetaan UML-koostesuhteena.
- Paikkatietokohteen ominaisuudet mallinnetaan luokan UML-attribuutteina.
- Jos ominaisuudella on itsellään omia ominaisuuksia, se mallinnetaan luokkana.
- Paikkatietokohteeseen liittyvät operaatiot mallinnetaan vastaavan luokan UML-operaatioina (huom. operaatioita ei yleensä mallinneta tiedonsiirtoon liittyvää sovelluskeemaa tehtäessä).
- Käytettävien nimien tulisi kuvata mahdollisimman tarkasti mallinnettavaa kohdetta tai ilmiötä.
- Kun nimi muodostuu useasta sanasta, aloitetaan jokainen sana ensimmäisen sanan jälkeen isolla kirjaimella ja lisäksi kaikki sanat kirjoitetaan yhteen ilman välilyöntejä.
- Luokkien ja pakettien nimet aloitetaan isolla kirjaimella.
 - Esimerkki luokan nimestä: VaikeakulkuinenSuo.
- Attribuuttien, operaatioiden ja suhderoolien nimet aloitetaan pienellä kirjaimella.
 - Esimerkki attribuutin nimestä: keskimaarainenPituus.
- Luokkien nimien tulee olla yksikäsitteisiä ja jokaisen luokan tulee kuulua johonkin pakettiin.

5 Tiedonsiirron sovelluskeemaan liittyvät lisäohjeet

Tiedonsiirtoon tarkoitettua sovelluskeemaa laadittaessa tulee noudattaa seuraavia ohjeita, jotka pohjautuvat *SFS-EN ISO 19136* -standardiin (GML).

- Laadittavien UML-mallien tulee olla täydellisiä, eli sisältää mallin kannalta oleelliset luokat, attribuutit, suhteet ja tietotyyppimääritykset.
- Suhteita, joissa on osallisena useampia kuin kaksi luokkaa, tulee välttää.
- Ainakin yksi suhderooli tulee osoittaa jokaista suhdetta kohti ja myös suhteiden toistuvuusmääreet tulee antaa.
- Attribuutit tulee määritellä näkyviksi luokan ulkopuolella.
- Kaikkien luokkanimien tulee olla yksikäsitteisiä ja soveltua XML-elementtinimiksi. Luokkien ja niiden attribuuttien nimissä ei tule käyttää skandinaavisia merkkejä, eikä alaviivaa, pistettä, väliviivaa tai muuta aakkosten ulkopuolista merkkiä.
- Kaikkien kohdeluokkaa edustavien UML-luokkien tulee olla varustettu stereotyyppillä `<<featureType>>`.
- Objektiluokat mallinnetaan UML-luokkina, joille ei anneta stereotyyppiä (geometriaobjektit, topologiaobjektit ja referenssijärjestelmät ovat esimerkkejä objektiluokista).
- Abstraktina kantaluokkana voidaan aina käyttää UML-luokkaa varustettuna stereotyyppillä `<<type>>`; tästä luokasta voidaan periyttää sekä kohdeluokkia, objektiluokkia että tietotyyppejä.
- Tietotyyppejä voidaan mallintaa UML-luokkina stereotyyppillä `<<enumeration>>`, `<<codeList>>`, `<<union>>` ja `<<dataType>>`.
- Kaikilla muilla paitsi objektiluokilla tulee olla niiden luonnetta osoittava stereotyyppi.
- Perintäsuhde voidaan luoda kahden luokan välille vain, jos ne molemmat edustavat joko kohdetyyppiä, objektityyppiä tai datatyyppiä.
- Moniperintää ei saa käyttää.
- Kaikilla attribuuteilla tulee olla nimi ja tietotyyppi.

JUHTA - Julkisen hallinnon tietohallinnon neuvottelukunta

- Attribuuttinimien on sovelluttava XML-elementtinimiksi.
- Suhteiden mahdolliset suunnat tulee osoittaa antamalla vastaava suhderoolinimi.
- Sovellusskeemaa vastaava UML-paketti varustetaan stereotyyppillä <<applicationSchema>>.

Paikkatietojen UML-mallinnuksessa suositellaan käytettävän taulukossa 1 listattuja attribuuttien tietotyyppejä. Tietotyyppien yhdenmukainen soveltaminen edistää tietomallien siirrettävyyttä ja oikeaa tulkintaa. Tietotyyppien yksityiskohtainen merkitys käy ilmi standardista *ISO/TS 19103:2002 Conceptual schema language (CSL)*.

Perustietotyypit	Koostetyypit
<ul style="list-style-type: none"> • Numeric <ul style="list-style-type: none"> - Decimal - Vector - Real - Number - UnlimitedInteger - Integer • Date and Time <ul style="list-style-type: none"> - DateTime - Date - DatePrecision - Time • Text <ul style="list-style-type: none"> - CharacterString - Sequence <Character> - Character - CharacterSetCode - LanguageCharacterString • Enumerations <ul style="list-style-type: none"> - Sign - Digit - Bit • Truth <ul style="list-style-type: none"> - Boolean - Logical - Truth - DiscreteTruth - ContinuousTruth - Probability • Multiplicities <ul style="list-style-type: none"> - Multiplicity - MultiplicityRange 	<ul style="list-style-type: none"> • Set • Bag • Sequence • Dictionary <p>Luetellut tyypit</p> <ul style="list-style-type: none"> • enumeration • codeList <p>Johdetut tyypit</p> <ul style="list-style-type: none"> • Measure • UnitOfMeasure • Area • UomArea • Length • Distance • UomLength • Angle • UomAngle • Scale • UomScale • Mtime • UomTime • Volume • UomVolume • Velocity • UomVelocity <p>Tyhjät tyypit</p> <ul style="list-style-type: none"> • NULL • EMPTY

Taulukko 1 Paikkatietojen UML-mallinnuksessa käytettäviksi suositellut tietotyypit (ISO/TS 19103, luku 6.5 Data types)

JUHTA - Julkisen hallinnon tietohallinnon neuvottelukunta

Paikkatietojen mallinnuksessa suositellaan käytettäväksi seuraavia UML-luokkien stereotyyppijä (*ISO/TS 19103* luvut 6.8 ja D.8.3, *EN ISO 19136* luvut 5.5 ja E.2):

- applicationSchema
- featureType
- dataType
- enumeration
- codeList
- union
- primitiveType
- leaf.