

CASE: Miten omia domain- spesifisiä kieliä tehdään?

Tekniseen
mallinrakentamiseen
liittyvää teoriaa ja
käytäntöä

Mikko Laakso

Sandvik Mining and
Construction Oy

mikko.laakso@sandvik.com

Taustaa

- Diplomityö Sandvik Mining and Construction Oy, System Platforms. 2008. "Distributed System Design Flow: Fielbus Modeling"
- 09/2008- Järjestelmäsuunnittelija, Competence Center Control Systems

Sisältö

- Mikä on mallinnuskieli (DSL, DSML)
- Mallinnusprosessista
- Mallinrakentamisen strategioita
- Mallinnusdemo

Mikä on domain-spesifinen kieli

- Domain-spesifinen mallinnuskieli (DSL, DSML) kuvaa muunneltavuuden
 - Vrt. platform
- Yksi määritelmä: formaali tietorakenteen kuvaus, jonka instanssia voidaan jalostaa
 - DSML on graafinen DSL
- Wikipedia: *"In software development, a domain-specific language (DSL) is a programming language or specification language dedicated to a particular problem domain, a particular problem representation technique, and/or a particular solution technique."*
 - Lyhenneviidakko: DSL, DSML, MDD, MDA, MDE, oAW, EMP, EMF, GMF, GEF, GEMS, GME, ...

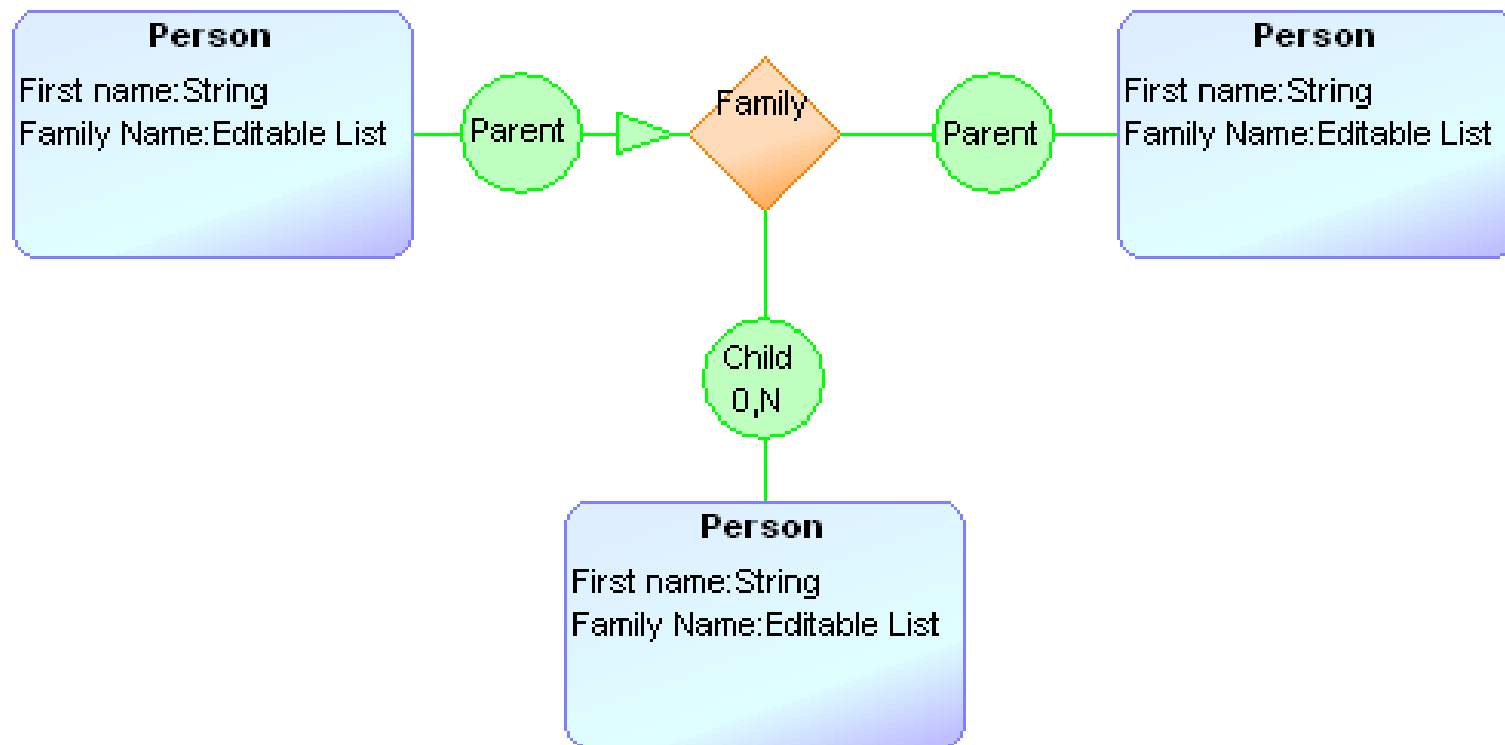
Mikä on domain-spesifinen kieli

- Kieleen kuuluu kielioppi eli metamalli, sen esitysmuoto, generaattorit sekä muut työkalut
 - Potentiaalisia metamalleja: esim.
 - XML-tiedoston rakenne (skeema)
 - Simulink lohkokaavio
 - C++
 - Regular expressions
 - Web-sivusto
 - Esitysmuotoja: Teksti, lohkokaavio, UML-diagrammi ja sen näkymät, traktorin tuoteoptiokuva, ...
 - Työkaluja: Metaedit+, Eclipse Modeling Tools, ...

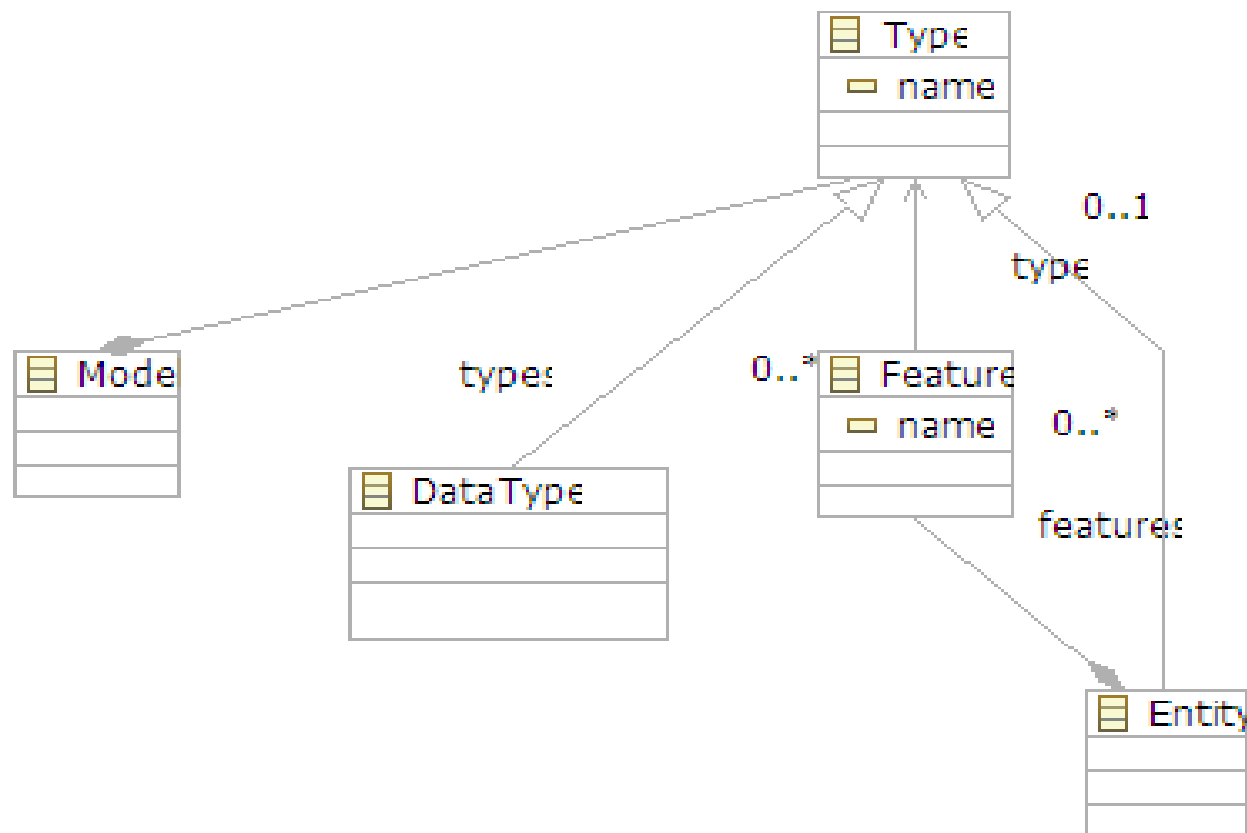
Mikä on domain-spesifinen kieli

- Rakentuu yleensä toisen mallikielen päälle
 - ”Metamallinnuskieli”: esim. GOPPRR, Ecore, ...
- Tuloksena ”Metamalli”, kielioppi joka kuvaa säännösten jonka avulla malli-instanssia voidaan rakentaa

Mikä on domain-spesifinen kieli



Mikä on domain-spesifinen kieli



Mikä on domain-spesifinen kieli

```
Model:
  (types+=Type)*;

Type:
  DataType | Entity;

DataType:
  "datatype" name=ID;

Entity:
  "entity" name=ID "{"
    (features+=Feature)*
  "}";

Feature:
  type=[Type | ID] name=ID;
```

Työkaluja eri käyttökohteisiin

- Graafinen mallinnus: Metaedit+ (, GEMS)
- XML-editorin luominen skeemasta (rajoituksin): Eclipse Modeling Framework (EMF)
- Tekstuaalisen metamallin luominen, editorin generointi: XText (oAW tai TMF)

Miten domain-spesifisiä kieliä voi käyttää

- Kun halutaan luoda ja verifioida jalostettavaa tietoa, esim. ohjelmointikieli, konfiguraatitiedosto
- Kun halutaan abstrahoida näennäinen monimutkaisuus
- Kun halutaan automatisoida tiedon generointi, esim. järjestelmän dokumentaatio
- Kun halutaan luoda uusi malli rakenteisen tiedon jäsentämiseen
- Kun ei ole valmista työkalua, tai kun halutaan rakentaa työkalu itse
- ...

Käyttökohteita

- Sovellusohjelman abstraktion graafinen mallinnus parametroimalla
- Tuoterakenteen muuntelun graafinen mallinnus,
- Tekstimuotoisen laitekonfiguraation tuottaminen graafisesta mallista
- XML-editorin generointi,
- ASCII-pohjaisen tiedostoformaatin editorin generointi generaattoreineen
- ...

Case: Graafisen mallikielen luominen

Käsitteitä

- Työkalut: Mallintamiseen liittyvät työkalut
 - metamallinnuskieli,
 - graafinen sovelluskehys,
 - generaattorit,
 - mallien syntaksin verifiointi
 - ...
- Mallinnusprosessi: mallin luominen työkaluja käyttämällä

Case: Graafisen mallikielen luominen

Mallinnusprosessi, roolit

- Mallintaja
 - Tuntee työkalut (konsultti, firman oma)
- Domain-ekspertti
 - Tuntee käyttöalueen
- Käyttäjä
 - Kuka työkalua käyttää?

Case: Graafisen mallikielen luominen

Mallinnusprosessi

- Kielen suunnittelu alkoi mallinnettavan asian rajaamisella
 - Mitä mallinnetaan
 - Mitä mallista halutaan tuottaa
 - Rajausta tarkennettiin myöhemmin lisää

Case: Graafisen mallikielen luominen

Mallinnusprosessi

- Domain-asiantuntijoiden säännölliset haastattelut
 - Mallinnettavan kohteen ymmärtäminen, mallin koheesio
 - Mallinnuksen tavoitteet
 - Mallin rakenteet
 - Iteratiivinen prosessi

Case: Graafisen mallikielen luominen

Mallinnusprosessi

- Mallin rakentaminen
 - Ecore-metamalli
 - Mallin näkymän/näkymien suunnittelu
 - Muunneltavuusnäkömä, mallinnusperusteet
 - Staattisuus/dynaamisuus
 - Laajuus
 - Laajennettavuus

Case: Graafisen mallikielen luominen

Mallinnusprosessi

- Datan generointi
 - Kun malli kypsynyt riittävästi
 - Malli mahdollistaa tiedossa olevan lopputuloksen
 - Casessa: Tuotettavat tiedostot ja niiden tietorakenteet
 - Generointitemplatet (XPand-templatekieli)
 - Generoitava data
 - Datan synteesi, muunnos uuteen muotoon

Case: Graafisen mallikielen luominen

Mallinnusprosessi

- Graafisen ulkoasun suunnittelu
 - Malli ei taipunut suoraan graafiseen esitykseen
 - Editori
 - Eclipse Graphical Modeling Framework (GMF) vaatii mallirakenteen mappauksen graafisiin elementteihin
 - Työläs ja hankala vaihe, jos GMF ei ole ollut mukana jo mallia rakennettaessa
 - Nykyään olemassa hieman parempia apuvälineitä, wizardeja jne
 - Tulossa: GEMS

Case: Graafisen mallikielen luominen

Mallinnusprosessi

- Kertaus:
 1. Domaintuntemus: Lähtötiedot → mitä halutaan
 2. Mallikielen luominen
 3. Generaattoreiden luominen

Mallin hallinnointi

- Yleensä mallinnuskieli sekä malli-instanssi versioituvat, generoidut asiat eivät

Graafinen mallinnus

- Demo: audiosignaalin reititys
 - Domain-spesifinen mallinnuskieli
 - Generaattori
 - XML-tiedoston generointi

Tekstuaalinen mallinnus

- XText (openArchitectureWare)
 - Java bean-generointi

Kysymyksiä?

- Kävimme läpi
 - Mallinnuksesta yleensä
 - Mallinnuscasen
 - Esimerkkimalleja

Kiitos!

Mikko Laakso

Sandvik Mining and Construction Oy

mikko.laakso@sandvik.com/

mikko.o.laakso@gmail.com

Puh. 0400 147 292