

**KAUNO TECHNOLOGIJOS UNIVERSITETAS**  
**PROGRAMŲ INŽINERIJOS KATEDRA**

**PROGRAMINĖS ĮRANGOS ĮDIEGIMO TYRIMAS**

Variantas 2.0

Vadovas:

Robertas Damaševičius

2011-01-24

Užsakovas:

Robertas Damaševičius

2011-01-24

Autorius:

IFM-9/2 gr.stud.

Kęstutis Matonis

2011-01-24

KAUNAS, 2011

# Turinys

|  |    |
|--|----|
| 1 PROJEKTO PARAIŠKA.....                                       | 6  |
| Taikymo sritis.....  | 6  |
| Projekto tikslas (Projekto vadovas).....                       | 6  |
| Projekto adresas (Projekto vadovas).....                       | 6  |
| Informacija apie užsakovo organizaciją (Projekto vadovas)..... | 6  |
| Problemos sprendimas pasaulyje.....                            | 7  |
| 1.1.1.1 Feature IDE [1].....                                   | 7  |
| 1.1.1.2 Xfeature [2].....                                      | 7  |
| 1.1.1.3 Feature modeling plugin [3].....                       | 7  |
| 1.1.1.4 Ecore Feature modeling plugin [4].....                 | 7  |
| 1.1.1.5 CaptainFeature [5].....                                | 8  |
| 1.1.1.6 RequiLine [6].....                                     | 8  |
| 1.1.1.7 FAMA [7].....  | 8  |
| 1.1.1.8 Gears [8].....   | 8  |
| 1.1.1.9 KumbangTools [9].....                                  | 8  |
| Produkto apibūdinimas.....                                     | 9  |
| Programų sistemos funkcijos.....                               | 9  |
| Sistemos kontekstas.....                                       | 9  |
| Vartotojo charakteristikos.....                                | 9  |
| Vartotojo problemos.....                                       | 9  |
| Vartotojo tikslai.....   | 9  |
| Bendri apribojimai.....  | 9  |
| projekto įgyvendinimo planai ir Kokybės vertinimas.....        | 9  |
| Projektą įgyvendinantis personalas.....                        | 9  |
| Projekto įgyvendinimo fazės.....                               | 10 |
| 1.1.1.10 Projekto biudžetas.....                               | 10 |
| 1.2 Išvados.....   | 10 |
| 2 PROJEKTAVIMO METODOLOGIJOS IR TECHNOLOGIJŲ ANALIZĖ.....      | 10 |
| 2.1 Įvadas.....  | 10 |
| 2.1.1 Požymių diagramos.....                                   | 10 |

|  |    |
|--|----|
| 2.1.2 Tekstinio požymių diagramų specifikavimo pagrindimas .....                     | 11 |
| 2.2 Žinomų XML grįstų specifikavimo kalbų analizė.....                               | 12 |
| 2.2.1 XMI (XML Metadata Interchange).....  | 12 |
| 2.2.1.1 Trumpas kalbos ir jos sintaksės bei semantikos aprašymas.....                | 12 |
| 2.2.1.2 Kalbos palaikomi standartai.....   | 12 |
| 2.2.2 OWL (Web Ontology Language).....   | 13 |
| 2.2.2.1 Trumpas kalbos ir jos sintaksės bei semantikos aprašymas.....                | 13 |
| 2.2.2.2 Kalbos palaikomi standartai.....   | 13 |
| 2.2.2.3 Galimybė aprašyti požymių diagramos elementus.....                           | 14 |
| 2.2.3 EBNF (Extended Bakus-Naur Form).....   | 15 |
| 2.2.3.1 Trumpas kalbos ir jos sintaksės bei semantikos aprašymas.....                | 15 |
| 2.2.3.2 Kalbos palaikomi standartai.....   | 16 |
| 2.2.3.3 Galimybė aprašyti požymių diagramos elementus.....                           | 16 |
| 2.3 Tipinės požymių diagramos specifikavimo analizuojamomis kalbomis pavyzdžiai..... | 17 |
| 2.3.1 Analizuotų kalbų įvertinimas.....  | 18 |
| 3 PROJEKTO PLANAS.....   | 19 |
| 4 ARCHITEKTŪROS SPECIFIKACIJA.....   | 20 |
| Architektūros tikslai ir apribojimai .....   | 20 |
| Panaudojimo atveju aprašymai .....   | 21 |
| Sistemos statinis vaizdas.....   | 22 |
| Eclipses platformos įskiepių apžvalga.....   | 23 |
| 5 TESTAVIMO MEDŽIAGA.....  | 24 |
| 5.1 Testavimo planas.....  | 24 |
| 5.1.1 Įžanga.....  | 25 |
| 5.1.1.1 Tikslai ir objektai .....  | 25 |
| 5.1.1.2 Terminų žodynas .....  | 25 |
| 5.1.1.3 Testavimo Technologijos .....  | 25 |
| 5.1.1.4 Reikalavimai testavimui .....  | 25 |
| 5.1.1.5 Testavimo prioritetai .....  | 26 |
| 5.1.1.6 Rolės ir atsakomybės .....   | 26 |
| 5.1.1.7 Testavimo Aplinka .....  | 26 |

|   |    |
|---|----|
| 5.1.1.7.1 Techninė įranga .....   | 26 |
| 5.1.1.7.2 Kliento programinė įranga, Operacinė sistema .....                      | 26 |
| 5.1.2 Specifiniai reikalavimai:.....  | 27 |
| 5.1.2.1 Testavimo Strategijos .....   | 27 |
| 5.1.2.2 Vartotojo sąsajos testavimas .....  | 27 |
| 5.1.2.3 Sistemos veikimo testavimas .....   | 28 |
| 5.1.2.4 Streso Testavimas .....   | 28 |
| 5.2 Testų aprašymas .....   | 29 |
| 5.2.1 Vartotojo sąsajos testavimas.....   | 29 |
| 5.2.1.1 Projekto sukūrimas.....   | 29 |
| 5.2.1.2 Diagramos elemento įtraukimas į diagramą .....                            | 30 |
| 5.2.1.3 Ryšio tarp diagramos elementų įtraukimas.....                             | 30 |
| 5.2.1.4 Diagramos elemento priskyrimas UML diagramos elementui .....              | 31 |
| 5.2.1.5 Diagramos išsaugojimas XML formatu .....                                  | 31 |
| 5.2.1.6 Diagramos išsaugojimas OWL formatu.....                                   | 32 |
| 5.2.1.7 Diagramos išsaugojimas FDL formatu.....                                   | 32 |
| 5.2.1.8 Diagramos išsaugojimas paveikslėlio formatu.....                          | 33 |
| 5.2.1.9 UML diagramos generavimas .....   | 34 |
| 5.2.1.10 Streso testavimas.....   | 34 |
| 5.2.1.11 Testavimo rezultatų kaupimas.....  | 34 |
| 6 LICENZIJA.....  | 35 |
| Eclipse Public License - v 1.0.....   | 35 |
| 7 VARTOTOJO DOKUMENTACIJA.....  | 40 |
| Požymių diagramų saugojimo formatų konvertavimo įrankio “fd2Convert” vadovas..... | 40 |
| Pavyzdys:.....  | 41 |
| Diagramų generavimas iš konvertuotų failų, naudojant FD2 sistemą.....             | 41 |
| Eclipse aplinka .....   | 42 |
| Naujos diagramos kūrimas .....  | 44 |
| Diagramos saugojimas į failą .....  | 47 |
| Failo atidarymas .....  | 48 |
| c)Galimos klaidos.....  | 48 |

|   |    |
|---|----|
| Instaliavimo vadovas .....                      | 49 |
| Sistemos reikalavimai .....                     | 49 |
| 7.1.1.1 Minimalūs kompiuterio reikalavimai..... | 49 |
| 7.1.1.2 Rekomenduojami reikalavimai.....        | 49 |
| Diegimo žingsniai .....                         | 50 |
| 7.1.1.3 Windows operacinėje sistemoje .....     | 50 |
| 7.1.1.4 Linux operacinėse sistemose.....        | 52 |
| 8 PAKEITIMŲ SĄRAŠAS.....                        | 52 |
| 9 LITERATŪRA.....                               | 52 |
| 10 TERMINŲ IR SANTRUMPŲ ŽODYNAS.....            | 53 |

# 1 PROJEKTO PARAIŠKA

## Taikymo sritis

### Projekto tikslas (*Projekto vadovas*)

SDarbo tikslas išanalizuoti požymių diagramų saugojimo formatus, ir juos pritaikyti plėtojant kuriamą programinę įrangą.

Pastaruoju metu programų produktų linijos inžinerija skatina sisteminių pakartotinių panaudojimą visame programos kūrimo procese. Programų produktų linijos pakartotinis panaudojimas stipriai priklauso nuo variantiškumo. Variantiškumui atvaizduoti yra naudojamos kelios technikos, viena iš jų yra požymiai. Projekto reikalavimus formavo užsakovas Robertas Damaševičius. Projektą įgyvendina magistrantas Kęstutis Matonis.

Projekto įgyvendinimo terminas 2010 09 15 - 2011 06 01 (18 mėn).

### Projekto adresas (*Projekto vadovas*)

Informatikos fakultetas, Programų inžinerijos katedra, Automatizuoto projektavimo procesų mokslinė grupė, Studentų 50-415, Vadovas: Robertas Damaševičius, email: [damarobe@soften.ktu.lt](mailto:damarobe@soften.ktu.lt)

### Informacija apie užsakovo organizaciją (*Projekto vadovas*)

Automatizuoto projektavimo procesų mokslinė grupė atlieka mokslinius tyrimus pagal tokią tematiką:

1) Aukšto lygmens specifikacijų ir reikalavimų bei variantiškumo (t.y., aukšto lygmens modelių) atvaizdavimas įvairių sistemų ar jų komponentų kūrime naudojant požymių diagramas ir požymiais grįsto modeliavimo koncepcijas.

2) Aukšto lygmens modelių transformavimas į meta-programas (generatyvinius komponentus) ir į meta-meta-programas, kad būtų galima pagerinti projektavimo proceso kokybę ir užtikrinti aukštesnį produktyvumą (našumą).

3) Programų transformavimo procesai (adaptavimas, modifikavimas, specializavimas, apgrąžos inžinerija) tikslu padidinti programų pakartotinio panaudojimo laipsnį, kokybę ir kitas charakteristikas.

## **Problemos sprendimas pasaulyje**

Žemiau pateikiami panašūs sprendimai.

### **1.1.1.1      *Feature IDE [1]***

Ši programa palaiko programų šeimynų kūrimą, pagrįstą AHEAD architektūros modeliu, kuris yra skirtas programų šeimynų kūrimui. AHEAD architektūroje laikoma, kad požymis, tai nedidelė sudedamoji sistemos dalis (klasė ar funkcija), kuri papildo programos funkcionalumą. Šis įrankis leidžia braižyti požymių diagramas ir kurti programinį kodą, pagal sukurtas požymių diagramas.

Šiuo metu šis įrankis yra kuriamas Magdeburgo (Vokietija) universitete.

### **1.1.1.2      *Xfeature [2]***

Tai požymių modeliavimo įrankis, kuris palaiko programų šeimynų modeliavimą. Šis įrankis leidžia vartotojams susikurti savo požymių meta modelį. Šis įrankis yra Eclipse programos priedas. Ši programa skirta pademonstruoti, kaip galima automatizuoti programų modeliavimą ir konfigūraciją pasitelkiant pakartotinai panaudojamas programos dalis.

Įrankį sukūrė Ondrej Rohlik ir Alessandro Pasetti, šiuo metu jis naudojamas bei tobulinamas [ETH-Zürich](#) (Automatikos institutas, Šveicarija).

### **1.1.1.3      *Feature modeling plugin [3]***

Šis įrankis yra Eclipse priedas, jis gali būti naudojamas vienas (leidžia modeliuoti požymių diagramas), arba kartu su fmp2rsm priedu, kuris sujungia šį įrankį su Rational Software Modeler (RSM) arba Rational Software Architect (RSA) modeliavimo įrankiais (leidžia modeliuoti programų šeimynų UML diagramas). Šis įrankis skirtas modeliuoti reikalavimams produktų šeimai ir aprašyti priklausomybes tarp požymių. Savybės vaizduojamos hierarchiškai. Grafiniam vaizdavimui pasitelkiamas medžio vaizdavimo principas.

Šį įrankį sukūrė Kanados Waterloo universitetas ir šiuo metu jo kūrimas nebetęsiamas.

### **1.1.1.4      *Ecore Feature modeling plugin [4]***

Tai [Eclipse](#) priedas leidžiantis grafiškai modeliuoti požymių diagramas. Jis leidžia peržiūrėti bei modeliuoti [EMF Ecore](#) (Eclipse modeline framework) modelius. Šis įrankis susieja

ECORE modelius taip, kad pakeitus Ecore modeli, pakeitimai matomi požymių diagramose ir atvirkščiai. Požymiai yra susiejami su Ecore modeliais.

Šiuo metu šis įrankis yra kuriamas Kanados Waterloo universitete (Nėra jokios versijos 2008-10-04).

#### **1.1.1.5      *CaptainFeature [5]***

Ši programa skirta programų šeimynų modeliavimo analizei. Šis įrankis neišbaigtas ir trūksta dokumentacijos, kad būtų aišku, kaip galima panaudoti šį įrankį. Šio įrankio neįtrauksime į tolesnius palyginimus

#### **1.1.1.6      *RequiLine [6]***

Tai reikalavimų surinkimo įrankis, sukurtas efektyviam produktų linijos valdymui. Šis įrankis leidžia modeliuoti produktus pasitelkiant požymių diagramomis ir reikalavimais. Iš sudarytų diagramų sukurama produkto konfigūracija. Šis įrankis turi grafini požymių diagramų redaktorių leidžianti dirbti su FORM notacijos diagramomis, Klaidų tikrinimą, XML redaktorių.

#### **1.1.1.7      *FAMA [7]***

Tai įrankis skirtas automatizuoti kintančių modelių analizavimą, panaudojant požymių modelius. Šis įrankis sudarytas iš kelių dalių: FAMA programa, FAMA karkasas ir FAMA Eclipse priedas. FAMA karkasas yra pagrindinis programos komponentas realizuojantis programos logiką, o pati programa ir Eclipse priedas realizuoja požymių diagramų vaizdavimą.

#### **1.1.1.8      *Gears [8]***

Gears yra programų šeimynos projektavimo įrankis ir turi skeletą skirtą projektuoti programinę įrangą, kuri skirtųsi tam tikrai požymiais. Programa sujungia skirtingus projektus į vieną ir leidžia juos valdyti kaip vieną, taip pateikiant visus programinės įrangos produktus, kaip vieną gaminį, tik skirtingomis atmainomis.

Šis įrankis yra komercinis ir nepavyko surinkti pakankamai informacijos apie jį, taigi informacija apie jį gali būti nepilna.

#### **1.1.1.9      *KumbangTools [9]***

Tai yra įrankių rinkinys konfigūruojamai programų šeimynai valdyti. Šiuo metu šį produktą sudaro šie



įrankiai: „Kumbang Modeller“ tai įrankis programų šeimynai projektuoti; „Kumbang Configurator“ tai įrankis programų šeimynos architektūros modelių kūrimui.

## **Produkto apibūdinimas**

### **Programų sistemos funkcijos**

Sistema padeda modeliuoti ir atvaizduoja požymių diagramomis

### **Sistemos kontekstas**

Produktas – programinė įranga.

### **Vartotojo charakteristikos**

Produktas skirtas mokslininkams. Vartotojas turi suprasti programų (komponentų) šeimynų variantiškumo modeliavimo naudojant požymių diagramas principus.

### **Vartotojo problemos**

Kuriamas produktas vartotojams padės vizualiai projektuoti sistemas, atliks validavimą. Sumažės klaidų skaičius.

### **Vartotojo tikslai**

Padės greičiau ir tiksliau modeliuoti sistemas

### **Bendri apribojimai**

Produktas turi būti realizuotas JAVA kalba, kaip „ECLIPSE“ įterpinys. Tai leis produktui veikti praktiškai bet kurioje operacinėje sistemoje.

## **projekto įgyvendinimo planai ir Kokybės vertinimas**

### **Projektą įgyvendinantis personalas**

Tai nuolatinis darbuotojas visam projekto gyvavimo laikotarpiui.

Reikiama kvalifikacija:

Turi būti susipažinęs su naudojamomis technologijomis (JAVA, Eclipse įterpinių kūrimas) programuoti, susipažinęs su įvairiais tekstinio saugojimo formatais.

Atliekami darbai:

- sistemos tobulinimas, saugojimo formato tobulinimas
- (1) sistemos diegimas ir testavimas

## **Projekto įgyvendinimo fazės**

Prieš pradėdant projektuoti sistemą reiktų atlikti tyrimus tokiose srityse:

- tekstinių saugojimo formatų analizė
- požymių diagramų, požymių diagramų vaizdavimo analizė

### **1.1.1.10 Projekto biudžetas**

Kadangi projektas nereikalauja jokio papildomos techninės įrangos, projekto biudžetas yra skirtas laikas sistemos kūrimui

## **1.2 Išvados**

Kuriama sistema yra aktuali, jos integravimas su plačiausiai naudojama „ECLIPSE“ platforma žymiai sutrumpins kūrimo laiką. Sistemos naudojimo intuityvumas ir paprastumas žymiai palengvins vartotojų darbą. Tekstinis saugojimo formatas, leis pernešti diagramas tarp kitų įrankių.

# **2 PROJEKTAVIMO METODOLOGIJOS IR TECHNOLOGIJŲ ANALIZĖ**

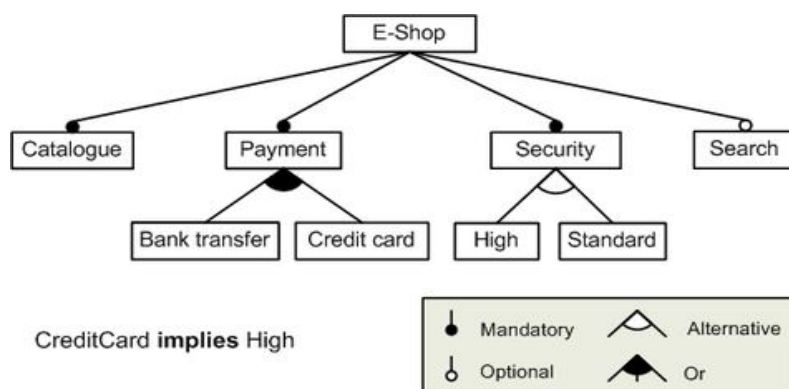
## **2.1 Įvadas**

### **2.1.1 Požymių diagramos**

Požymis tai savitas, charakteringas sistemos atributas. FODA (Feature Oriented Domain Analysis) [1] požymius apibūdina kaip žinomas, savitas bei vartotojui matomas sistemos charakteristikas, tuo tarpu funkcijos, objektai ir aspektai yra naudojami apibūdinti vidinės sistemos detales. Požymių modeliavimas susitelkia ties labai matomų išorinių produkto charakteristikų apibūdinimu, kalbant apie produkto bendrumą bei variantiškumą, o ne apie detalių sistemos apibūdinimą.[2]

Požymių modeliavimo rezultatas yra požymių diagramos. Tai yra grafinė kalba naudojama atvaizduoti bei modeliuoti sistemos arba komponento

variantiškumus aukštesniame abstrakcijos lygyje, dažniausiai pradiniuose projektavimo lygiuose, tokiuose kaip reikalavimų specifikavime kuriant programinę įrangą. [3] Požymiai yra apibūdinami ir klasifikuojami, galimybių specifikacijose, sričių technologijose, kūrimo technikose bei operacinėse aplinkose. Galimybės yra vartotojui matomos charakteristikos kurios gali būti apibūdinamos kaip skirtingos paslaugos, operacijos ir nefunkcinės charakteristikos. Sričių technologijos apibūdina paslaugų kūrimą arba operacijas. Kūrimo metodai yra bendros funkcijos arba metodai kurie naudojamos kurti paslaugas, operacijas ar srities funkcijas. Operacinės aplinkos apibūdina kurioje aplinkoje programinė įrangą yra naudojama.



## 1. Požymių diagramos pavyzdys [10]

### 2.1.2 Tekstinio požymių diagramų specifikavimo pagrindimas

Šiuo metu yra daug neatitikimų tarp įvairių modeliavimo programų naudojamų saugojimo formatų. Sistemos negali apsikeisti duomenimis, nėra būdo, kaip perkelti meta modelius iš vienos sistemos į kitą. [4]

Tekstinis diagramų saugojimo formatas, yra vienintelis būdas, leidžiantis perduoti meta duomenis iš vienos sistemos į kitą. Todėl kuriama programinė įrangą, būtinai turi specifiuoti požymių diagramas tekstiniu formatu.

## **2.2 Žinomų XML grįstų specifikuojamų kalbų analizė.**

### **2.2.1 XMI (XML Metadata Interchange)**

#### **2.2.1.1 Trumpas kalbos ir jos sintaksės bei semantikos aprašymas.**

**XML Metadata Interchange (XMI)** yra Object Management Group (OMG) standartas duomenų apsiuvinėjimui panaudojant XML kalbą.[5]

Ji gali būti naudojama išreikšti bet kuriu meta duomenis, kurių meta modelis gali būti išreikštas Meta-Object Facility (MOF).

#### **2.2.1.2 Kalbos palaikomi standartai.**

XMI integruoja keturis industrijos standartus [5]:

- XML - eXtensible Markup Language, W3C standartas.
- UML - Unified Modeling Language, OMG modeliavimo standartas.
- MOF - Meta Object Facility, OMG language for specifying metamodels.
- MOF Mapping to XMI

## 2.2.2 OWL (Web Ontology Language)

### 2.2.2.1 Trumpas kalbos ir jos sintaksės bei semantikos aprašymas.

1 lentelė - OWL kalbos sintaksė ir semantika[6,7]

| OWL  | Shared informal semantics                          |
|--|--|
| rdf:Property<br>owl:DatatypeProperty<br>owl:ObjectProperty | Named relation between nodes or nodes and values   |
| rdfs:subPropertyOf   | Relation can appear in place of a more general one |
| rdfs:range   | The relation range kind                            |
| owl:Class  | Relations and contextual restrictions package      |
| owl:Restriction  | Contextualised restriction of a relation           |
| rdfs:subClassOf  | Package concretises the base package               |
| owl:maxCardinality<br>owl:minCardinality                   | Restrict the number of occurrences of a relation   |
| owl:intersectionOf<br>owl:unionOf                          | Combination of relations in a context              |

### 2.2.2.2 Kalbos palaikomi standartai.

OWL naudoja šiuos standartus[7]:

- XML - eXtensible Markup Language, W3C standartas.
- RDF - Resource Description Framework

### 2.2.2.3 Galimybė aprašyti požymių diagramos elementus.

2 lentelė - Požymių diagramos specifikavimas naudojant OWL [8]

| Elementas                          | Apibrėžimas   | Grafinis pavyzdys | OWL aprašymas   |
|------------------------------------|---|-------------------|---|
| Požymis                            |   |                   | <code>&lt;owl:Class rdf:ID="A"/&gt;</code>  |
| Būtinasis                          | Požymis B (C,D) yra įtrauktas, jeigu jo tėvas A yra įtrauktas<br>Jei A tuomet B;<br>Jei A tuomet B ir C   |                   | <code>&lt;owl:Class rdf:ID="A"/&gt;</code><br><code>&lt;owl:intersectionOf</code><br><code>  rdf:parseType="Collection"&gt;</code><br><code>    &lt;owl:Class rdf:ID="C" /&gt;</code><br><code>    &lt;owl:Class rdf:ID="D" /&gt;</code><br><code>&lt;/owl: intersectionOf&gt;</code><br><code>&lt;/owl:Class&gt;</code>                            |
| Nebūtinasis                        | Požymis B(C,D) gali būti įtrauktas jeigu jo tėvas A yra įtrauktas<br>Jei A tuomet B arba niekas;<br>Jei A tuomet C arba D arba niekas   |                   | <code>&lt;owl:Class rdf:ID="A"/&gt;</code><br><code>&lt;owl:unionOf</code><br><code>  rdf:parseType="Collection"&gt;</code><br><code>    &lt;owl:Class rdf:ID="C" /&gt;</code><br><code>    &lt;owl:Class rdf:ID="D" /&gt;</code><br><code>    &lt;owl:Nothing /&gt;</code><br><code>&lt;/owl:unionOf&gt;</code><br><code>&lt;/owl:Class&gt;</code> |
| Pasirenkamas (case - pasirinkimas) | Tik vienas požymis B arba C arba D turi būti pasirinktas jeigu jo tėvas A yra parinktas<br>Jeigu A tuomet case iš(B,C,D)  |                   | <code>&lt;owl:Class rdf:ID="A"&gt;</code><br><code>&lt;owl:oneOf</code><br><code>  rdf:parseType="Collection"&gt;</code><br><code>    &lt;owl:Class rdf:ID="B"/&gt;</code><br><code>    &lt;owl:Class rdf:ID="C"/&gt;</code><br><code>&lt;/owl:oneOf&gt;</code><br><code>&lt;/owl:Class&gt;</code>  |
| Pasirenkamas (or - pasirinkimas)   | Mažiausiai vienas požymis (B,C arba D arba B ir C; arba B ir D; arba C ir D; arba B ir C ir D) turi būti pasirinkti jeigu jo tėvas A yra pasirenkamas<br>Jei A tuomet bet kuris iš(B,C,D) |                   | <code>&lt;owl:Class rdf:ID="A"/&gt;</code><br><code>&lt;owl:someValuesFrom</code><br><code>  rdf:parseType="Collection"&gt;</code><br><code>    &lt;owl:Class rdf:ID="B" /&gt;</code><br><code>    &lt;owl:Class rdf:ID="C" /&gt;</code><br><code>&lt;/owl:someValuesFrom&gt;</code><br><code>&lt;/owl:Class&gt;</code>                             |

|                                |   |  |   |
|--------------------------------|---|--|---|
| Ryšys<br>(xor)                 | Ryšys tar dviejų<br>atominių<br>požymių. K xor F.<br>Jei F tuomet ne K<br>ir jei ne F tuomet<br>K |  | <owl:Class rdf:ID="F"/><br><owl:complementOf><br><owl:Class rdf:ID="K"/><br></owl:complementOf><br></owl:Class> |
| Reikalavi<br>mas<br>(requires) | Požymis K<br>reikalauja<br>požymio F.<br>K reikalauja F   |  |   |
| Values                         |   |  | rdfs:Literal, rdfs:Number   |

## 2.2.3 EBNF (Extended Backus-Naur Form)

### 2.2.3.1 Trumpas kalbos ir jos sintaksės bei semantikos aprašymas.

**Extended Backus-Naur Form (EBNF)** yra šeima meta sintaksės žymėjimų, naudojamų išreikšti nuo konteksto nepriklausomai gramatikai. Tai formalus būdas išreikšti kompiuterių programavimo kalbas ar kitas formalias kalbas. Tai Backus-Naur Form (BNF) meta sintaksės žymėjimo praplėtimas. [9]

3 lentelė - EBNF kalbos sintaksė ir semantika[10]

| BNF   | Reikšmė         |
|---|-----------------|
| ::= concept-name details<br>{ feature }   | Požymių modelis |
| ::= feature-name [details]  | Požymis         |
| ::= "(" feature-list ")"   //<br>required features<br>  "[" feature-list "]"   // optional<br>features                              | Detalės         |
| ::= element { " " element }   //<br>one only<br>  element { "+" element }   //<br>one or more<br>  element { "," element }   // all | Požymių sąrašas |
| ::= feature<br>  details  | Elementas       |

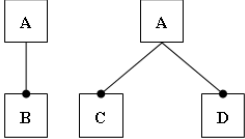
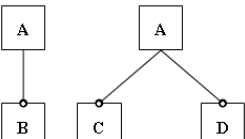
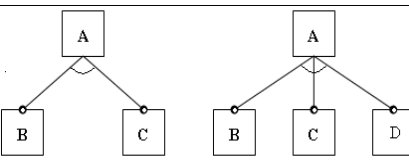
### 2.2.3.2 Kalbos palaikomi standartai.

EBNF naudoja šiuos standartus:

- BNF (Backus–Naur Form) (išplečia)

### 2.2.3.3 Galimybė aprašyti požymių diagramos elementus.

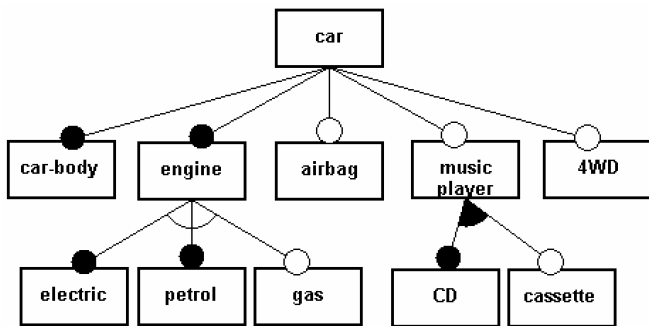
4 lentelė - Požymių diagramos specifikavimas naudojant BNF

| Elementas                          | Apibrėžimas   | Grafinis pavyzdys   | OWL aprašymas    |
|------------------------------------|---|---|------------------|
| Požymis                            |   |   | ::= feature-name |
| Būtinasis                          | Požymis B (C,D) yra įtrauktas, jeigu jo tėvas A yra įtrauktas<br>Jei A tuomet B;<br>Jei A tuomet B ir C                               |    | A ( B,C )        |
| Nebūtinasis                        | Požymis B(C,D) gali būti įtrauktas jeigu jo tėvas A yra įtrauktas<br>Jei A tuomet B arba niekas;<br>Jei A tuomet C arba D arba niekas |   | A [ C +D ]       |
| Pasirenkamas (case - pasirinkimas) | Tik vienas požymis B arba C arba D turi būti pasirinktas jeigu jo tėvas A yra parinktas<br>Jeigu A tuomet case iš(B,C,D)              |  | A ( B C D )      |



|                                   |   |  |                   |
|-----------------------------------|---|--|-------------------|
| Pasirenka mas (or - pasirinkimas) | Mažiausiai vienas požymis (B,C arba D arba B ir C; arba B ir D; arba C ir D; arba B ir C ir D) turi būti pasirinkti jeigu jo tėvas A yra pasirenkamas. Jei A tuomet bet kuris iš(B,C,D) |  | $A ( B + C + D )$ |
|-----------------------------------|---|--|-------------------|

## 2.3 Tipinės požymių diagramos specifikuojamomis kalbomis pavyzdžiai



### 2. Automobilio požymių diagrama

#### a) Požymių diagramos specifikuojimas naudojant OWL

```

<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF>
  <owl:Class rdf:ID="car">
    <owl:intersectionOf rdf:parseType="Collection">
      <owl:Class rdf:ID="car-body" />
      <owl:Class rdf:ID="engine" />
    </owl:intersectionOf>
    <owl:unionOf rdf:parseType="Collection">
      <owl:Class rdf:ID="airbag" />
      <owl:Class rdf:ID="music-player" />
      <owl:Class rdf:ID="4WD" />
    </owl:unionOf>
  </owl:Class>

```

```

<owl:Class rdf:ID="engine">
  <owl:oneOf rdf:parseType="Collection">
    <owl:Class rdf:ID="electric"/>
    <owl:Class rdf:ID="petrol"/>
    <owl:Class rdf:ID="gas"/>
  </owl:oneOf>
</owl:Class>
<owl:Class rdf:ID="music-player">
  <owl:someValuesFrom rdf:parseType="Collection">
    <owl:Class rdf:ID="electric"/>
    <owl:Class rdf:ID="petrol"/>
    <owl:Class rdf:ID="gas"/>
  </owl:someValuesFrom>
</owl:Class>
</rdf:RDF>

```

#### b) Požymių diagramos specifikuojamas naudojant BNF

```
car ( car-body, engine(electric|petrol|gas) + airbag + music-player[CD+cassette] + 4WD)
```

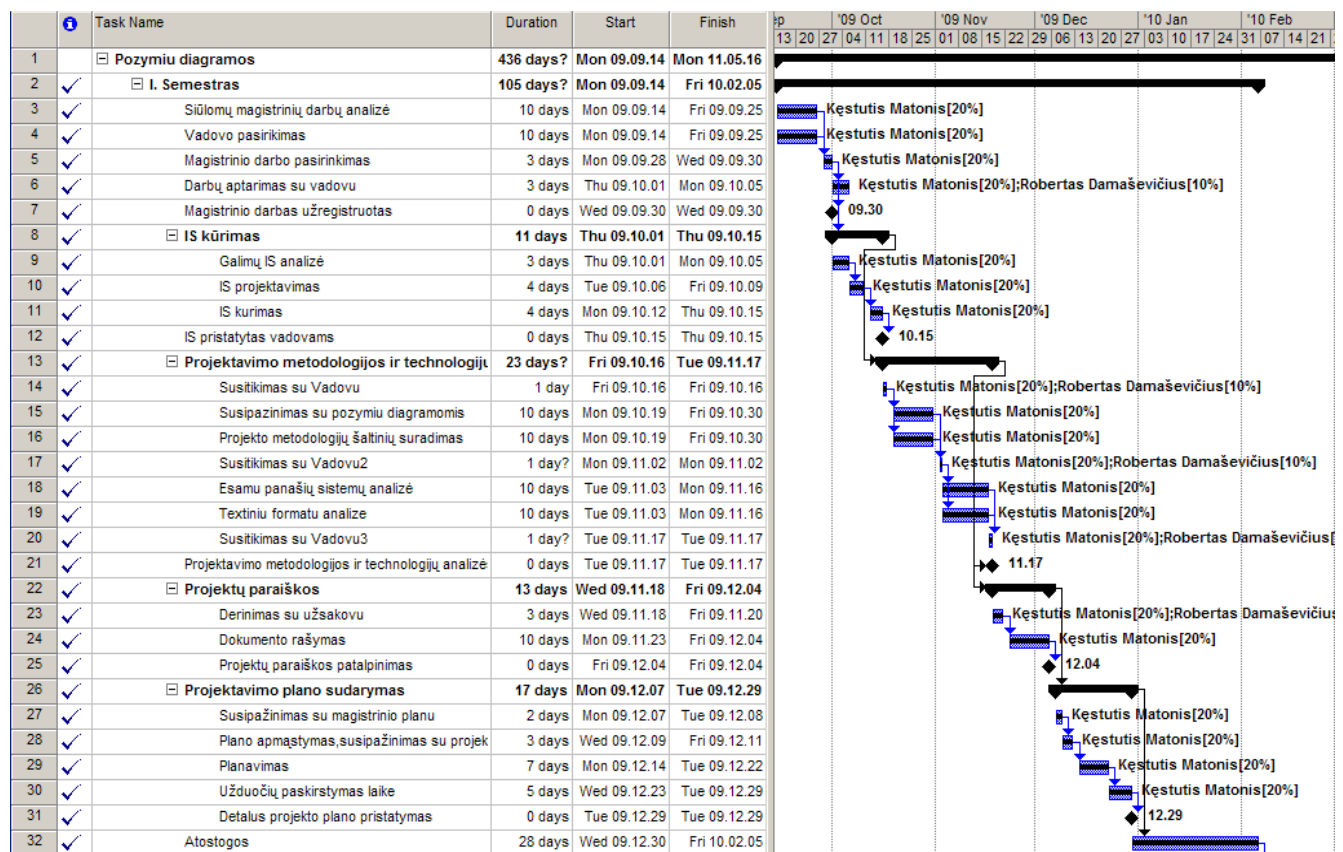
### 2.3.1 Analizuotų kalbų įvertinimas

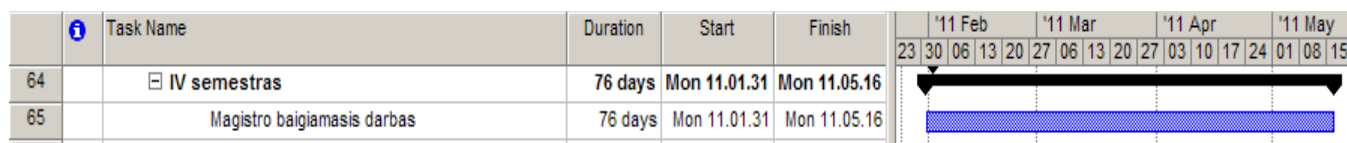
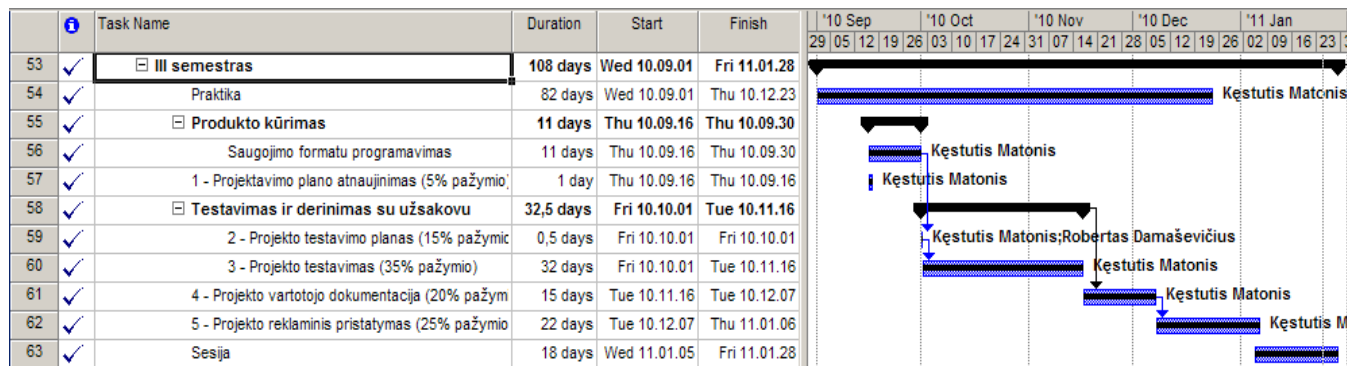
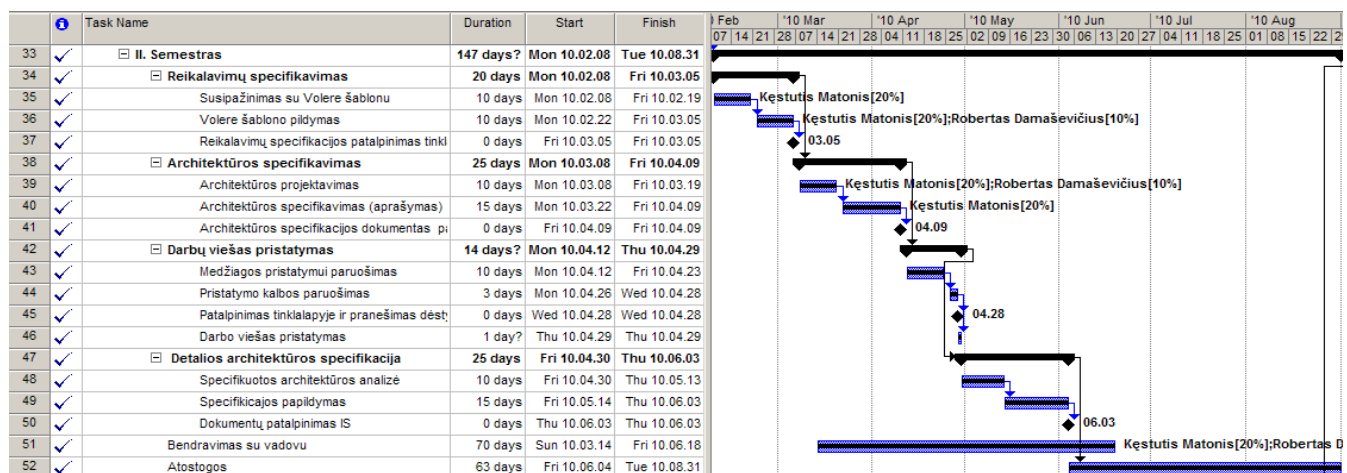
5 lentelė - Kalbų įvertinimas

| Kalba      | Pranašumai   | Trūkumai  |
|------------|--|---|
| <b>XMI</b> | <ul style="list-style-type: none"> <li>naudoja pripažintą XML užrašymo standartą, kuris labai palengvina duomenų apsikeitimą</li> <li>galima pilnai aprašyti požymių diagramas</li> <li>lanksti</li> </ul> | <ul style="list-style-type: none"> <li>sudėtinga</li> <li>perteklinė</li> <li><b>ilgas aprašymas</b></li> </ul> |
| <b>OWL</b> | <ul style="list-style-type: none"> <li>naudoja pripažintą XML užrašymo standartą, kuris labai palengvina duomenų apsikeitimą</li> <li>galima pilnai aprašyti</li> </ul>                                    | <ul style="list-style-type: none"> <li>sudėtinga</li> <li>perteklinė</li> <li>ilgas aprašymas</li> </ul>        |

|      |   |   |
|------|---|---|
|      | požymių diagramas <ul style="list-style-type: none"> <li>lanksti</li> </ul>   |   |
| EBNF | <ul style="list-style-type: none"> <li>Lengvai aprašoma</li> <li>Galima pilnai aprašyti požymių diagramas</li> <li>lengvai suprantama</li> <li>trumpas aprašas</li> </ul> | 1. negalima naudoti duomenų apsikeitimui tarp kitų programų<br>2. nelanksti |

### 3 PROJEKTO PLANAS





## 4 ARCHITEKTŪROS SPECIFIKACIJA

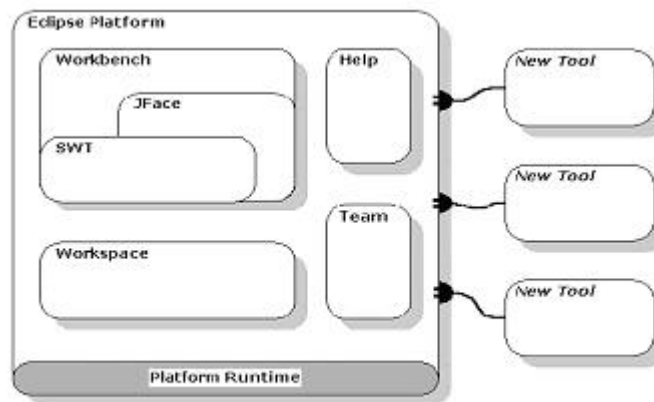
### Architektūros tikslai ir apribojimai

Šiuo metu nėra numatyta galimybė įdiegti kuriamą priedą į kitas programas, todėl apsiribojame, kad priedas veiks tik Eclipse platformoje.

Eclipse programa yra parašyta JAVA programavimo kalba. Ši programavimo kalba yra objektinė programavimo kalba ir tinkama akademinių projektų įgyvendinimui. Todėl siekiant išvengti

suderinamumo su kitomis programavimo kalbomis klaidų, priedas bus įgyvendintas taip pat JAVA programavimo kalba.

Mūsų kuriamo priedo architektūra yra įtakojama programos, kuriai jis bus kuriamas. Šiuo atveju tai yra Eclipse. Eclipse tai programa, kuri pagaminta pagal specialią architektūrą, kuri leidžia, priedų pagalba, praplėsti jos veikimą. Viena Eclipse programa be priedų nebūtų taikomoji programa - visas jos veikimas pagrįstas jai parašytų priedų veikimu. Taigi priedo architektūra bus pritaikyta darbui su Eclipse aplinka.



1 paveikslukas. Eclipse – priedų platforma. [<http://www.ibm.com>, 2009-04]

Taip pat mūsų priedas bus gaminamas pasitelkiant EMF bei GMF įrankius, kurie sukurs jiems būdingą programos architektūrą. Mūsų tikslas sukurti tinkamas duomenų struktūras, kurios leistų išvengti atsitiktinių klaidų.

GMF (the Graphical Modeling Framework) tai įrankių rinkinys, skirtas kurti modeliavimo įrankius, tokius kaip UML redaktoriai, duomenų srautų diagramas, ir panašiai.

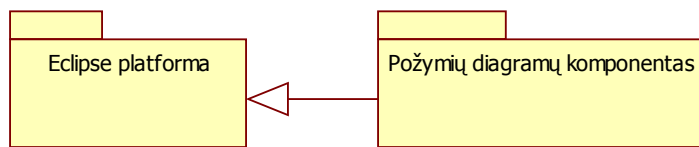
Mūsų kuriamas Eclipse komponentas turi sietis su UML modeliavimo komponentu Eclipse sistemoje. Tam pasirenkamas UML2tools Eclipse komponentas. Šis komponentas dirba su naujausia UML 2.2 sintakse ir palaiko bazines diagramas.

## Panaudojimo atveju aprašymai

|                |  |
|----------------|--|
| • Skaitymas iš | <b>Tikslas:</b> Atidaromas diagramos failas.<br><b>Aktoriai:</b> Užsakovas |
|----------------|--|

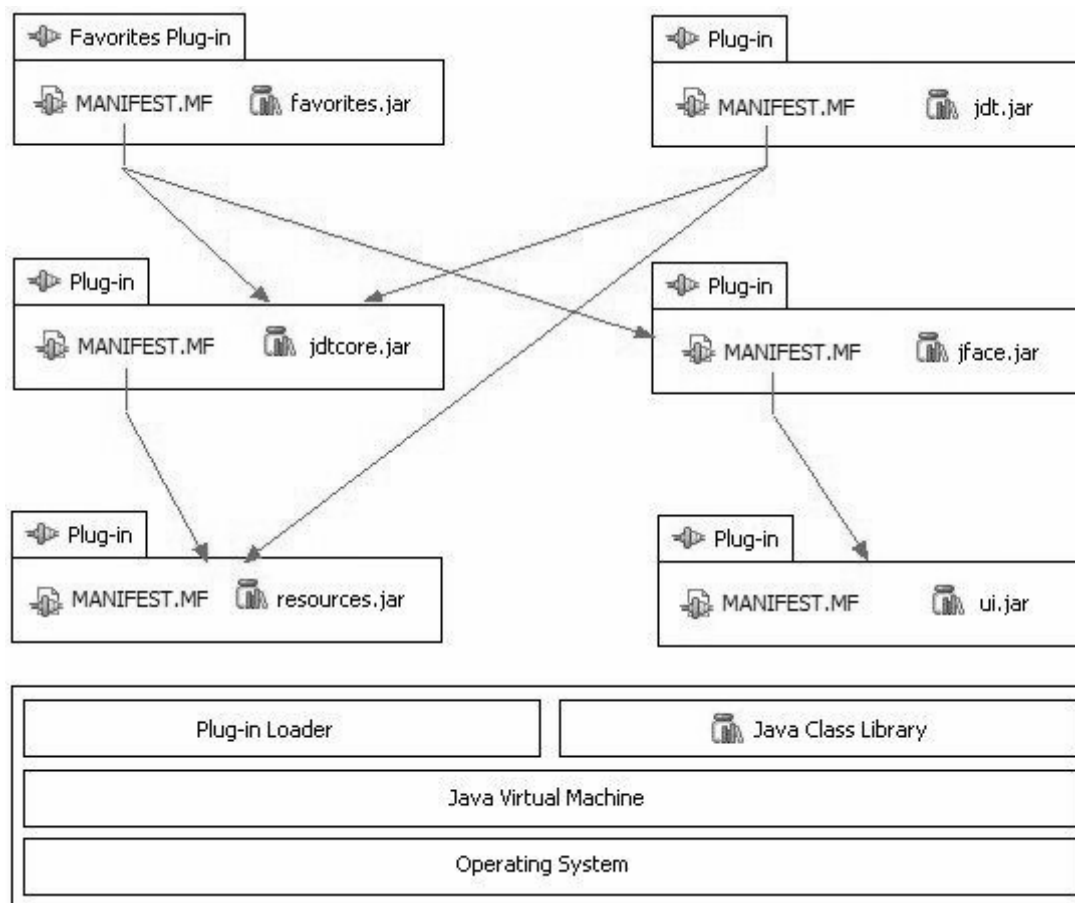
|  |   |
|--|---|
| failo  | <p><b>Ryšiai su kitais PA:</b> -</p> <p><b>Nefunkciniai reikalavimai:</b> -</p> <p><b>Prieš sąlygos:</b> -</p> <p><b>Sužadinimo sąlyga:</b> Spaudžiamas meniu mygtukas „atidaryti“.</p> <p><b>Po sąlyga:</b> Atidaroma diagrama.</p> <p><b>Pagrindinis scenarijus:</b> Spaudžiamas meniu mygtukas „atidaryti“, nurodomas diagramos failas.</p> <p><b>Alternatyvūs scenarijai:</b> -</p> <p><b>Architektūriniai ypatumai:</b> Skaitymui bus panaudotos standartines JAVA programavimo kalbos bibliotekos, galinčios apdoroti XML failus. Nuskaityti duomenys bus saugomi kompiuterio atmintyje, individualiose duomenų struktūrose. Tada duomenys atvaizduojami grafiškai.</p>   |
| <ul style="list-style-type: none"> <li>Saugojimas į failą</li> </ul> | <p><b>Tikslas:</b> Saugoma atidaryta diagrama į failą.</p> <p><b>Aktoriai:</b> Užsakovas</p> <p><b>Ryšiai su kitais PA:</b> -</p> <p><b>Nefunkciniai reikalavimai:</b> -</p> <p><b>Prieš sąlygos:</b> Atidaryta diagrama.</p> <p><b>Sužadinimo sąlyga:</b> Spaudžiamas meniu mygtukas „saugoti“.</p> <p><b>Po sąlyga:</b> Diagrama išsaugoma kietame diske.</p> <p><b>Pagrindinis scenarijus:</b> Spaudžiamas meniu mygtukas „saugoti“, nurodoma vieta ir failo pavadinimas.</p> <p><b>Alternatyvūs scenarijai:</b> Galima atšaukti procesą.</p> <p><b>Architektūriniai ypatumai:</b> Iš individualių duomenų struktūrų, duomenys transformuojami ir išsaugomi XML faile. XML failų rašymui bus panaudotos standartinės JAVA programavimo kalbos bibliotekos.</p> |

### Sistemos statinis vaizdas



Statinis sistemos vaizdas.

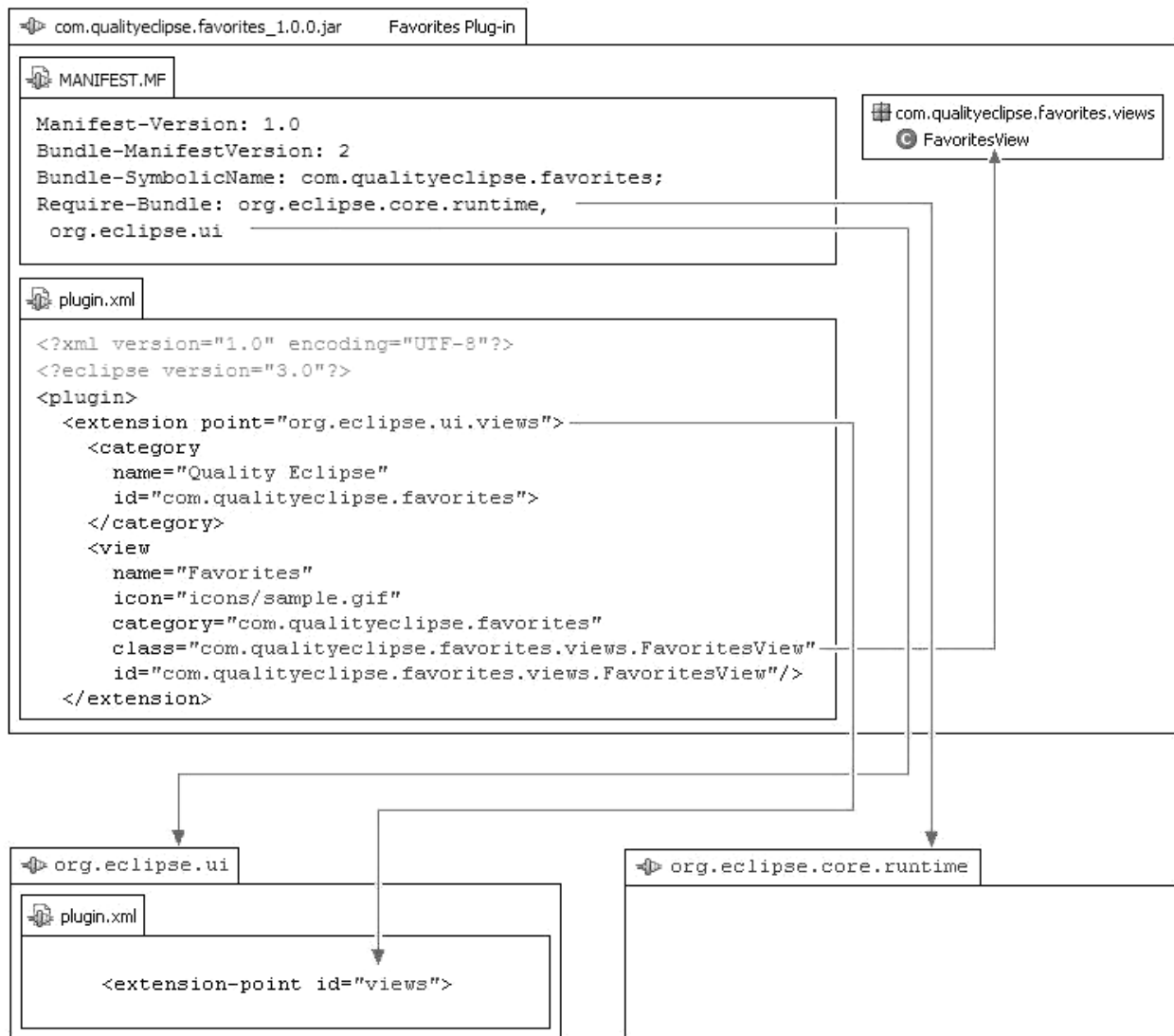
## Eclipse platformos įskiepių apžvalga



2 paveikslukas. Eclipse aplinkoje, daugelis komponentų yra priklausomi vienas nuo kito. [Eclipse plug-ins, Third Edition, 2009]

Eclipse programavimo aplinka sudaryta iš daugybės vienas nuo kito priklausančių komponentų. Kiekvieno komponento veikimas yra įgyvendintas realizacijoje (programavimo kalbos kode), tačiau nuo ko priklausomas ir ką atlieka komponentas yra aprašoma „MANIFEST.MF“ ir „plugin.xml“ failuose. Šis išdėstymas naudojamas sėkmingai įgyvendinti uždelstą komponentų paleidimą (lazy-loading) programoje, tai reiškia, kad kai kurie komponentai nėra įjungti, nors programa ir paleista. Taip sutaupoma sistemos resursų ir pati programa veikia kur kas greičiau. Eclipse programos paleidimo metu yra nuskaitomi visi „MANIFEST.MF“ ir „plugin.xml“ failai, ir esant reikalui, užkraunamas reikiamas komponentas į atmintį.

Pateikiame pavyzdį, kaip gali atrodyti komponentas įjungtas į Eclipse sistemą, kaip komponentas nurodo ryšius į kitus komponentus



3 paveiksliukas. Komponento aprašymas Eclipse programoje. [Eclipse plug-ins, Third Editon, 2009]

## 5 TESTAVIMO MEDŽIAGA

### 5.1 Testavimo planas



### 5.1.1 Įžanga

Šiame skyriuje yra pateikta apibendrinta požymių diagramų modeliavimo įrankio plėtinio programinės įrangos testų specifikacija.

#### 5.1.1.1 Tikslai ir objektai

Pateikti vartotojui teisingai funkcionuojantį produktą.

Šiame dokumente pateikiamai testavimo atvejai, padedantys užtikrinti aukštą įrankio programinės įrangos kokybę ir minimalų defektų kiekį.

#### 5.1.1.2 Terminų žodynas

lentelė .1. Terminų žodynas

| Terminas  | Apibūdinimas  |
|---|---|
| Eclipse programavimo aplinka su modeliavimo įrankių komplektu | Programa, priedų platforma, kurią galima papildyti individualiais priedais. |
| XML   | Extensible Markup Language  |
| FDL   | Feature Description Language  |
| OWL   | Web Ontology Language   |

#### 5.1.1.3 Testavimo Technologijos

Požymių diagramų plėtinio komponentui kokybės užtikrinimui bus atlikti:

- Naudotojo sąsajos testavimas;
- Komponento testavimas;
- Integracijos tarp komponentų testavimas;
- Visos sistemos testavimas;
- Sistemos pertestavimas po klaidų pataisymo.

#### 5.1.1.4 Reikalavimai testavimui

Norint atlikti programinės įrangos testavimą reikalinga:

- Pilna programinės įrangos specifikacija išreikšta panaudojimo atvejų diagramomis ir panaudojimo scenarijais.
- Pilna ir testavimui parengta programinė įranga.
- Nustatyta procedūra, ką daryti su aptiktomis klaidomis.

- Klaidos ataskaitos šablonas.

#### **5.1.1.5 Testavimo prioritetai**

Atliekant sistemos testavimą pagrindinis dėmesys bus kreipiamas į šiuos faktorius. Jie pateikti svarbumo mažėjimo tvarka.

- Funkcionalumas – ar reikalingos programinės įrangos funkcijos yra ir atliekam joms priskirtas roles taip kaip reikia (atitinka specifikaciją)
- Panaudojamumas – kaip vartotoja sąsaja yra patogi vartotojui ir draugiška.
- Saugumas – ar programa ir vartotojas gali sugadinti pradinius duomenis.

#### **5.1.1.6 Rolės ir atsakomybės**

Atliekant programinės įrangos testavimą yra apibrėžtos tokios rolės ir jų atsakomybės:

- Vadovas – atsakingas už testavimo planus ir sėkmingą jų vykdymą.
- Testuotojas – atsakingas už testų vykdymą pagal sudarytą planą.
- Produkto vadovas – užtikrina, kad testai yra įvykdyti sėkmingai žiūrint iš vartotojo perspektyvos.

#### **5.1.1.7 Testavimo Aplinka**

##### **5.1.1.7.1 Techninė įranga**

Testavimas turi būti atliekamas naudojant techninę įrangą, kuri atitiktų reikalavimus:

- 1 paveikslas.      Personalinio kompiuterio  
rekomenduojama minimali specifikacija: 1.5 GHz  
spartos procesorius, 512MB darbinės atminties,
- 2 paveikslas.      Rezoliucija ne mažesnė nei 1024 x 768.

##### **5.1.1.7.2 Kliento programinė įranga, Operacinė sistema**

Komponentas turi veikti operacinėse sistemose:

- Windows XP
- Windows 2000 profesional/server
- Windows 2003

- Windows Vista
- Linux

## 5.1.2 Specifiniai reikalavimai:

(2) Įrankis turi būti įkomponuotas į Eclipse Modeling Tools

### 5.1.2.1 Testavimo Strategijos

Kuriant testavimo atvejus ir testuojant Požymių diagramų braižymo komponentą turi būti naudojamos šios strategijos:

- ▲ Vartotojo sąsajos testavimas;
- ▲ Streso testavimas.
- ▲ Vienetų testavimas
- ▲ Integravimo testavimas
- ▲ Priemimo testavimas
- ▲ Aukšto lygio testavimas

### 5.1.2.2 Vartotojo sąsajos testavimas

Vartotojo sąsajos testavimo metu yra tikrinama ar vartotojas gali pasiekti visas reikiamas funkcijas. Tikrinama ar UI yra intuityvi..

|                              |  |
|------------------------------|--|
| <b>Testo tikslas:</b>        | Patikrinti ar visi vartotojo interfeiso komponentai yra pasiekiami.<br>Ar galima pereiti nuo lango prie lango, nuo lauko prie lauko naudojantis klaviatūra, pelės pagalba ir sparciaisiais mygtukais.<br>Visi langų objektai, tokie kaip meniu, objektų dydžiai, vieta lange ir kiti, atitinka numatytus standartus. |
| <b>Technologija:</b>         | Sukurti ir įvykdyti testavimo atvejus kiekvienam sistemos langui ir visiems esantiems objektams jame.  |
| <b>Užbaigimo kriterijai:</b> | Vartotojas gali exportuoti diagramas į visus saugojimo formatus  |

Kuriant vartotojo sąsajos testavimo atvejį reikia numatyti tokią informaciją:

- [1] *Testavimo atvejo pavadinimas*. Pavadinimą rekomenduojama rašyti tokį patį kaip ir testuojamo lango pavadinimas.

[2] *Testavimo atvejo scenarijus*. Testavimo scenarijų rekomenduojama rašyti lentelės forma.

Žemiau nurodyta lentelėje turi būti pateikiami testavimo atvejai:

| Nr | Objektas | Testavimo sąlygos, atvejis | Laukiami rezultatai |
|----|----------|----------------------------|---------------------|
|    |          |                            |                     |

#### 5.1.2.3 Sistemos veikimo testavimas

Sistemos veikimo testas yra naudojamas, norint nustatyti ir įvertinti funkcijų vykdymo laikus, reakcijos greičius ir kitus skaičiuojamus parametrus. Šio testo tikslas yra patikrinti ar sistema veikia pagal nustatytus sistemos veikimo reikalavimus. Veikimo testas turi būti parašytas ir įvykdytas pagal nurodytus sistemos veikimo techninius reikalavimus.

Žemiau nurodyta lentelėje turi būti pateikiami testavimo atvejai:

| Nr | Atvejis | Matavimo parametras |
|----|---------|---------------------|
|    |         |                     |

#### 5.1.2.4 Streso Testavimas

Streso testavimas yra vienas iš sistemos veikimo testų, kurio tikslas patikrinti sistemos veikimą esant mažiems resursams arba dirbant dideliu kiekiu sistemos naudotojų. Esant atminties ar vietos diske trūkumui gali būti aptinkamos klaidos, kurios nėra matomos normaliomis sąlygomis.

|                       |   |
|-----------------------|---|
| <b>Testo tikslas:</b> | <p>Patikrinti ar sistemos funkcijos teisingai ir be klaidų veikia esant stresinėms situacijoms:</p> <ul style="list-style-type: none"><li>• Esant mažai ar trūkstant kompiuterio atminties (RAM).</li><li>• Daug kartų kartojant nekorektiškas funkcijų veikimo situacijas bei jas keičiant.</li></ul> <p><b>Pastaba:</b> streso testo metu taip pat gali būti aprašomos situacijos, kurioms esant sistema veiks neteisingai.</p> |
|-----------------------|---|

|                              |   |
|------------------------------|---|
| <b>Technologija:</b>         | <ul style="list-style-type: none"> <li>• Panaudoti testavimo atvejus aprašytus veikimo ir apkrovos testuose.</li> <li>• Norint patikrinti sistemos veikimą esant mažiems kompiuterio resursams, testas gali būti vykdomas viename kompiuteryje.</li> <li>• Testuojant keliems sistemos naudotojams, testo duomenys gali būti naudojami tie patys. Šiuo atveju gali būti nustatomos situacijos, kuriose sistema veikia nekorektiškai.</li> </ul> |
| <b>Užbaigimo kriterijai:</b> | Testais yra skaitomas baigtas, kai visi suplanuoti testai yra suvykdyti ir aprašyti visi apribojimai sistemos veikimui (pateiktos situacijos, kurioms esant komponentas neveiks).   |

Kuriant naudotojo sąsajos testavimo atvejį reikia numatyti tokią informaciją:

- *Testavimo atvejo pavadinimas.*
- *Tikslas.* Trumpai aprašyti testavimo atvejo tikslą.
- *Pradinės sąlygos.* Aprašyti pradinę testavimo situaciją.
- *Testavimo atvejo scenarijus.* Testavimo scenarijų rekomenduojama rašyti lentelės forma.
- *Laukiami rezultatai.* Aprašyti laukiamą sistemos reakciją.

Žemiau nurodyta lentelėje turi būti pateikiami testavimo atvejai:

| Nr | Naudotojo veiksmas | Laukiami rezultatai |
|----|--------------------|---------------------|
|    |                    |                     |

## 5.2 Testų aprašymas

### 5.2.1 Vartotojo sąsajos testavimas

#### 5.2.1.1 Projekto sukūrimas

Tikslas: Sukurti naują požymių diagramą.

Pradinės sąlygos: Sistema prieinama vartotojui; Vartotojo techninė bei programinė įranga atitinka

reikalavimus

Susiję testavimo atvejai: Visi

Veikėjai: Visi

Scenarijus:

| Nr | Naudotojo veiksmas   | Laukiami rezultatai   |
|----|--|---|
| 1  | Pasileisti Eclipse programą  | Atidaroma programa ir užkraunamas požymių diagramos braižymo komponentas. |
| 2  | Atsidariusioje programoje susikurti standartinį projekto katalogą File->New Project -> General -> Project                                | Programos Project Explorer lange sukuriamas naujas projekto katalogas     |
| 3  | Spaudžiamas dešinysis pelės klavišas ant sukurto projekto katalogo ir jam priskiriama požymių diagrama New -> Exsample... -> Fd2 diagram | Sukuriama požymių braižymo diagrama ir atidaros braižymo langas           |

#### **5.2.1.2 Diagramos elemento įtraukimas į diagramą**

Tikslas: Įtraukti į diagramą naują požymių diagramos elementą.

Pradinės sąlygos: Sukurtas naujas projektas.

Susiję testavimo atvejai: Visi

Veikėjai: Visi

Scenarijus:

| Nr | Naudotojo veiksmas   | Laukiami rezultatai   |
|----|--|---|
| 1  | Išsirenkamas norimas diagramos elementas šoninėje įrankių juostoje ir pažymimas pele | Norimas įtraukti elementas nuspalvinamas oranžine spalva, taip parodant kad jis yra aktyvus |
| 2  | Diagramos brėžimo lange, norimoje vietoje spustelima pelės kairiuoju klavišu         | Naujas diagramos elementas atvaizduojamas diagramos brėžimo lange                           |
| 3  | Įvedamas elemento pavadinimas  | Požymių diagramos elementas atvaizduojamas su nauju pavadinimu.                             |

#### **5.2.1.3 Ryšio tarp diagramos elementų įtraukimas**

Tikslas: Sujungti du diagramos elementu ryšio elementu.

Pradinės sąlygos: Įterpti mažiausiai du diagramos elementai.

Susiję testavimo atvejai: Visi

Veikėjai: Visi

Scenarijus:

| Nr | Naudotojo veiksmas   | Laukiami rezultatai   |
|----|--|---|
| 1  | Išsirenkamas norimas ryšio elementas šoninėje įrankių juostoje ir pažymimas pele   | Norimas įtraukti ryšio elementas nuspalvinamas oranžine spalva, taip parodant kad jis yra aktyvus |
| 2  | Diagramos brėžimo lange pasirenkamas pirmas diagramos elementas ir paspaudžiamas, po to išsirenkamas antras elementas ir pažymimas | Tarp dviejų diagramos elementų nubrėžiamas ryšys  |

#### **5.2.1.4 Diagramos elemento priskyrimas UML diagramos elementui**

Tikslas: Priskirti UML diagramos elementą.

Pradinės sąlygos: Įterpti mažiausiai du diagramos elementai.

Susiję testavimo atvejai: Visi

Veikėjai: Visi

Scenarijus:

| Nr | Naudotojo veiksmas  | Laukiami rezultatai   |
|----|---|---|
| 1  | Išsirenkamas diagramos elementas, spaudžiamas dešinysis pelės klavišas ir pasirenkama priskirti UML diagramos elementą. | Atidaromas UML diagramos pasirinkimo langas   |
| 2  | Išsirenkama UML diagrama ir nurodomas elementas kuris bus priskirtas ir spaudžiama OK                                   | Prie diagramos elemento priskiriamas UML diagramos elementas ir atvaizduojamas paveikslukas šalia elemento. |

#### **5.2.1.5 Diagramos išsaugojimas XML formatu**

Tikslas: Išsaugoti požymių diagramą XML formatu.

Pradinės sąlygos: Nubrėžta požymių diagrama.

Susiję testavimo atvejai: Visi

Veikėjai: Visi

Scenarijus:

| Nr | Naudotojo veiksmas                    | Laukiami rezultatai                      |
|----|---------------------------------------|--|
| 1  | Pasirenkama diagrama Project Explorer | Atidaromas diagramos eksportavimo langas |

|   |  |   |
|---|--|---|
|   | lange ir spaudžiamas dešinysis pelės klavišas ir pasirenkama Export                      |   |
| 2 | Atsidariusiame lange pasirenkamas norimas eksportavimo formatas (XML) ir spaudžiama Next | Pasirenkamas XML saugojimo formatas         |
| 3 | Pasirenkama norima eksportuoti diagrama ir spaudžiama Next                               | Pasirenkama eksportuojama diagrama          |
| 4 | Atsidariusiame dokumento pavadinimo lange įvedamas norimas dokumento pavadinimas         | Sukuriamas norimu pavadinimu XML dokumentas |

#### **5.2.1.6 Diagramos išsaugojimas OWL formatu**

Tikslas: Išsaugoti požymių diagramą OWL formatu.

Pradinės sąlygos: Nubrėžta požymių diagrama.

Susiję testavimo atvejai: Visi

Veikėjai: Visi

Scenarijus:

| <b>Nr</b> | <b>Naudotojo veiksmas</b>   | <b>Laukiami rezultatai</b>                  |
|-----------|---|---|
| 1         | Pasirenkama diagrama Project Explorer lange ir spaudžiamas dešinysis pelės klavišas ir pasirenkama Export | Atidaromas diagramos eksportavimo langas    |
| 2         | Atsidariusiame lange pasirenkamas norimas eksportavimo formatas (OWL) ir spaudžiama Next                  | Pasirenkamas OWL saugojimo formatas         |
| 3         | Pasirenkama norima eksportuoti diagrama ir spaudžiama Next  | Pasirenkama eksportuojama diagrama          |
| 4         | Atsidariusiame dokumento pavadinimo lange įvedamas norimas dokumento pavadinimas                          | Sukuriamas norimu pavadinimu OWL dokumentas |

#### **5.2.1.7 Diagramos išsaugojimas FDL formatu**

Tikslas: Išsaugoti požymių diagramą FDL formatu.

Pradinės sąlygos: Nubrėžta požymių diagrama.

Susiję testavimo atvejai: Visi

Veikėjai: Visi



Scenarijus:

| Nr | Naudotojo veiksmas  | Laukiami rezultatai                         |
|----|---|---|
| 1  | Pasirenkama diagrama Project Explorer lange ir spaudžiamas dešinysis pelės klavišas ir pasirenkama Export | Atidaromas diagramos eksportavimo langas    |
| 2  | Atsidariusiame lange pasirenkamas norimas eksportavimo formatas (FDL) ir spaudžiama Next                  | Pasirenkamas FDL saugojimo formatas         |
| 3  | Pasirenkama norima eksportuoti diagrama ir spaudžiama Next  | Pasirenkama eksportuojama diagrama          |
| 4  | Atsidariusiame dokumento pavadinimo lange įvedamas norimas dokumento pavadinimas                          | Sukuriamas norimu pavadinimu FDL dokumentas |

### 5.2.1.8 Diagramos išsaugojimas paveikslėlio formatu

Tikslas: Išsaugoti požymių diagramą paveikslėlio formatu.

Pradinės sąlygos: Nubrėžta požymių diagrama.

Susiję testavimo atvejai: Visi

Veikėjai: Visi

Scenarijus A:

| Nr | Naudotojo veiksmas   | Laukiami rezultatai  |
|----|--|--|
| 1  | Diagramos lange spaudžiamas dešinysis pelės klavišas ir pasirenkama File -> Save As Image File.. | Atidaromas paveikslėlio saugojimo langas                               |
| 2  | Pasirenkama saugojimo vieta, formatas.   | Požymių diagrama išsaugojama pasirinktoje vietoje, pasirinktu formatu. |

Scenarijus B:

| Nr | Naudotojo veiksmas   | Laukiami rezultatai  |
|----|--|--|
| 1  | Diagramos lange spaudžiama dešiniuoju pelės klavišu diagramos kairiame viršutiniame kampe ir nepaleidžiant pelės klavišo vedama iki diagramos dešiniojo apatinio kampo ir klavišas atleidžiamas. | Pažymimi diagramos elementai kurie telpa į apvestą langą.  |
| 2  | Spaudžiamas dešinysis pelės klavišas ir pasirenkama Copy (alternatyva Ctrl+C)  | Diagrama nukopijuojama į atmintį ir naudojant įklįjavimo funkciją programose kurios priima vaizdinės informacijos įklįjavimą, galima įklįjuoti |

|  |  |   |
|--|--|---|
|  |  | pasirinkus reikiamą komandą (arba Ctrl+V) |
|--|--|---|

### **5.2.1.9 UML diagramos generavimas**

Tikslas: Sugeneruoti UML diagramą iš požymių diagramos.

Pradinės sąlygos: Nubrėžta požymių diagrama ir priskirti UML diagramos elementai.

Susiję testavimo atvejai: Visi

Veikėjai: Visi

Scenarijus:

| <b>Nr</b> | <b>Naudotojo veiksmas</b>   | <b>Laukiami rezultatai</b>                                     |
|-----------|---|--|
| 1         | Pasirenkama diagrama Project Explorer lange ir spaudžiamas dešinysis pelės klavišas ir pasirenkama Export | Atidaromas diagramos eksportavimo langas                       |
| 2         | Atsidariusiame lange pasirenkamas norimas eksportavimo formatas (UML) ir spaudžiama Next                  | Pasirenkamas UML saugojimo formatas                            |
| 3         | Pasirenkama norima eksportuoti diagrama ir spaudžiama Next  | Pasirenkama eksportuojama diagrama                             |
| 4         | Atsidariusiame dokumento pavadinimo lange įvedamas norimas dokumento pavadinimas                          | Sugeneruojama UML diagrama ir sukuriamas diagramos dokumentas. |

### **5.2.1.10 Streso testavimas**

Tikslas: Patikrinti sistemos veikimą sumažinus atminties kiekį (vykdomas testas LTC# 1)

Pradinės sąlygos: Sistema prieinama vartotojui; Vartotojo techninė bei programinė įranga atitinka reikalavimus; Vartotojas pasileidęs Eclipse sistemą su požymių braižymo komponentu.

Scenarijus: Vykdomas didelės požymių diagramos generavimas su minimaliais techniniais reikalavimais.

Laukiami rezultatai: Sugeneruota UML diagrama

### **5.2.1.11 Testavimo rezultatų kaupimas**

Testavimo pabaigoje tikimasi gauti šiuos dokumentus:

- Testavimo planas – šis dokumentas, taip pat jo pakeitimai, jei tokie bus.
- Pakeitimų reikalavimai – klaidų ataskaitos, reikalingų pakeitimų programinei įrangai

dokumentai.

- Užbaigimo ataskaita – dokumentas rodantis, kad buvo atliktas sėkmingai priėmimo testavimas

Testų rezultatai bus fiksuojami žemiau nurodytos lentelės formatu.

| Klaidos numeris | Klaidos pavadinimas ir aprašymas | Vieta programoje | Būsena |
|-----------------|----------------------------------|------------------|--------|
| 1               |                                  |                  |        |
| 2               |                                  |                  |        |
| ...             |                                  |                  |        |

## 6 LICENZIJA

### Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

#### DEFINITIONS

"Contribution" means:

- a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
- b) in the case of each subsequent Contributor:
  - i) changes to the Program, and
  - ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

## **GRANT OF RIGHTS**

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

## **REQUIREMENTS**

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

## **COMMERCIAL DISTRIBUTION**

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b)

allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

## **NO WARRANTY**

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

## **DISCLAIMER OF LIABILITY**

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial

in any resulting litigation.

## 7 VARTOTOJO DOKUMENTACIJA

### Požymių diagramų saugojimo formatų konvertavimo įrankio “fd2Convert” vadovas

Programos instaliuoti. Programa paleidžiama iš “jar” failo. Reikia nueiti į darbinę fd2Convert.jar direktoriją, ir paleisti programą.

Nenurodžius jokių parametrų, išvedama pagalba:

```
java -jar fd2Convert.jar
Požymių diagramų saugojimo formatų konvertavimo įrankis
Galimas daugkartinis konvertavimas panaudojant *.fdl|*.fd2|
Pagal numatytuosius nustatymus, transformuota kalba įrašoma į atitinkamą failą
Galimos operacijos (parašykite atitinkamą skaičių):
--help: išveda šia pagalbą, išveda jei nenurodomi jokie parametrai
      --fd2-fdl <nustatymai> <failas>: vidinio formato FD2 konvertavimas į FDL
      --fd2-owl <nustatymai> <failas>: vidinio formato FD2 konvertavimas į OWL
      --fdl-fd2 <nustatymai> <failas>: FDL formato konvertavimas į vidinį FD2
nustatymai:
  --i-konsolė: rezultatai bus išvesti į konsolę
  --nerasyti: rezultatai nebus rašomi į failą
```

Programa palaiko 3 krypčių tekstinių failų konvertavimą:

- Iš fd2 į fdl

iš vidinio FD2 sistemos formato, į lengvai žmogaus skaatomą, ir dėl to dažnai naudojamą fdl formatą (yra daug požymių diagramų išsaugotų šiame formate)

- Iš fd2 į owl

plačiai naudojamą WEB antologijų kalbą

- Iš fdl į fdi

konvertavimas atliekamas, siekiant suintegruoti konvertavimo įrankį su FD2 sistema. To dėka, galima generuoti požymių diagramų grafikus, naudojant FD2 iskiepį

Pagal nustatymus įrankis įrašo konvertuotą kalbą į atitinkamą konteinerį, ir neišveda teksto į konsolę (konvertuojan daug failų, rezultatų išvedimas žymiai sulėtina konvertavimo greitį).

Norint tai pakeisti, naudojami programos nustatymai:

- --nerasyti: gautas rezultatas **\*neišsaugomas\*** į failą
- --i-konsolė: gautas rezultatas išvedamas į konsolę

Įrankis buvo sukurtas, planuojant naudoti “batch” konvertavimą (visi failai atitinkantys tam tikrus



kriterijus, yra perduodama įrankiui, ir jis atlieką konvertavimą)

### **Pavyzdys:**

```
java -jar fd2Convert.jar --fd2-fdl *.fd2
```

visi failai turintys .fd2 galūnę, bus perduoti įrankiui, ir jis sugeneruos atitinkamus failus su .fdl galūne

### **Diagramų generavimas iš konvertuotų failų, naudojant FD2 sistemą**

Tam pirma reikia suinstaliuoti FD2 sistemą. Vadovas sekančiame skyrelyje.

Norint atvaizduoti požymių diagramas grafiškai, reikia turėti fd2 formatą tekstinį failą, ir užkrautą FD2 įskiepi.

Susikuriame paprastą projektą:

File->New->Project

General->Project

Sukuriame pvz\_diagrama

Į atsiradusio projekto direktoriją kopijuojam sugeneruotą shop.fd2 diagramą

Spaudžiame dešinį mygtuką ant sukurto projekto Eclipsės viduje

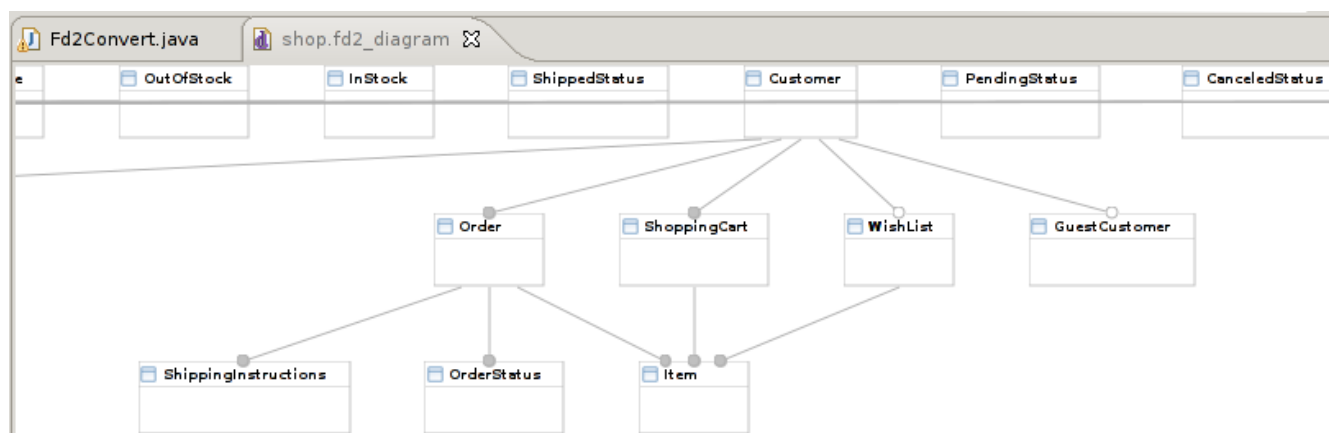
```
<projekto_direktorija> -> "Refresh"
```

Projekto vaizde atsiranda shop.fd2 failas

Spaudžiam dešinį mygtuką ant "shop.fd2" failo ir pasirenkam "Initialize fd2\_diagram file"

Atsiradusiame dialoge pasirenkame esamo projekto direktoriją ir spaudžiam "Finish"

Iškarto atsirado diagramų braižymo perspektyva su sugeneruota diagrama(diagrama matoma ne visą dėl didelio sudėtingumo):

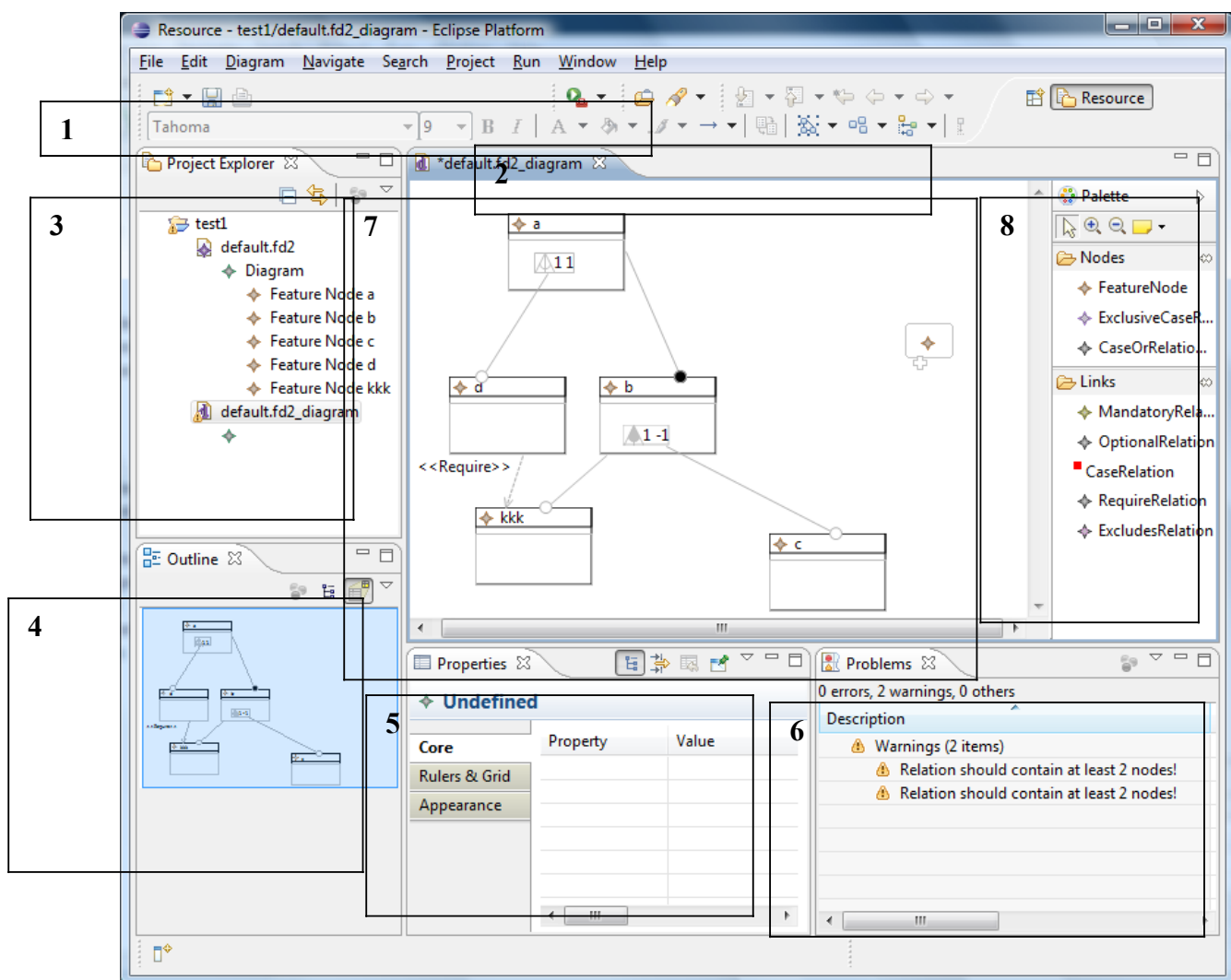


Ją galima pakeisti, ir po to vėl perkonvertuoti į fdl, ar owl formatą

## Eclipse aplinka

Standartinis Eclipse vaizdas

Pažymėti Eclipse programos komponentai trumpai aprašyti “Vartotojo sąsajos komponentai” lentelėje



3 paveikslas. Eclipse vartotojo sąsaja.

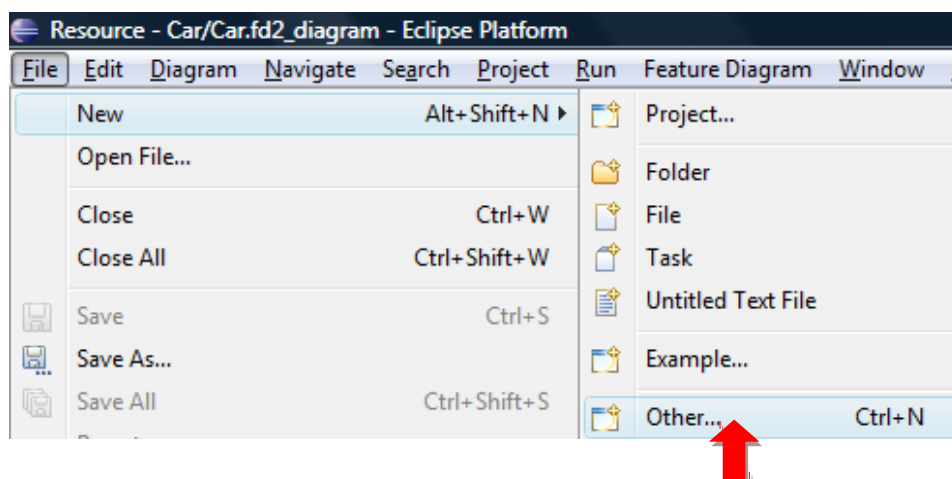
- Vartotojo sąsajos komponentai

| Komponento pavadinimas | Aprašymas  |
|------------------------|--|
| 1. Menu                | Meniu juosta. Šioje juostoje patalpinti Meniu pasirinkimai.                                    |
| 2. Toolbar             | Įrankių juosta.  |
| 3. Project Explorer    | Šiame laukelyje matomas modeliavimo projekto vaizdas. Modelių, diagramų, ryšių ir kiti failai. |
| 4. Outline             | Šis laukelis pateikia sumažinta vizualios diagramos vaizduojamos pagrindiniame lauke vaizdą.   |
| 5. Properties          | Pateikiamas diagramos elementų savybių sąrašas   |
| 6. Problems            | Jeigu diagramoje įvyksta klaidų, tai jų sąrašas pateikiamas                                    |

|                   |   |
|-------------------|---|
|                   | šiam laukelyje.   |
| 7. Diagram Window | Šis langas yra naudojamas diagramos vaizdavimui ir kūrimui. Jame vaizduojami grafiniai diagramos elementai ir ryšiai.   |
| 8. Tool box       | Šiame sąraše pateikiami įrankiai, kuriuos galima naudoti kuriant požymių diagramą. Šiuos įrankius galima nuvilkti tiesiai ant Diagramos Lango ir juos paleisti. Taip į diagramos langą bus įterpiami elementai. |

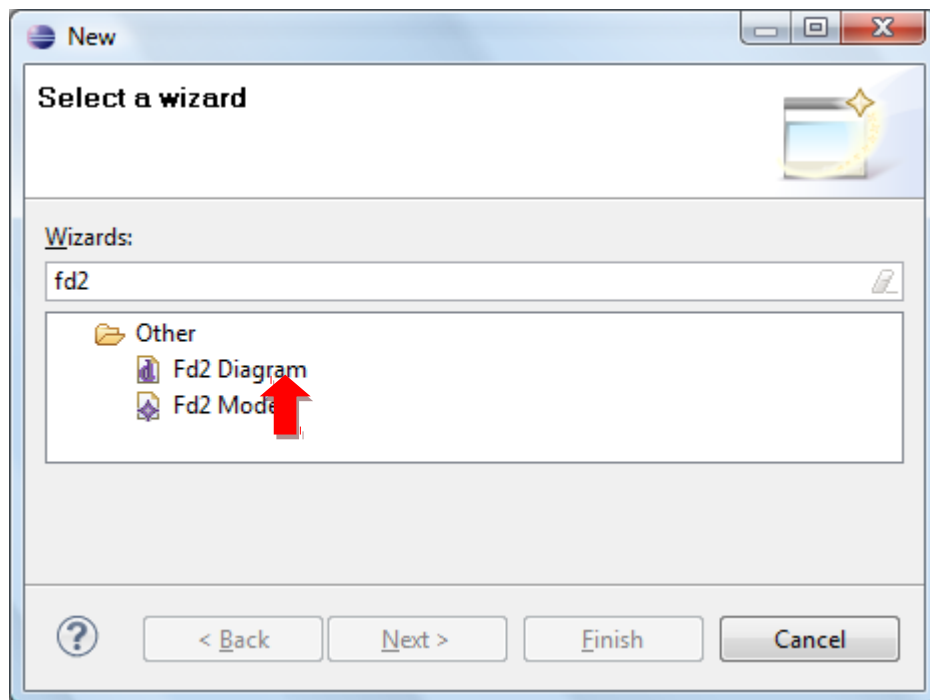
## Naujos diagramos kūrimas

Norėdami sukurti naują požymių modelį ar diagramą, Eclipse programavimo aplinkoje spauskite meniu punktą: „File -> New -> Other...“ arba naudokite greitąją klavišų kombinaciją „Ctrl + N“



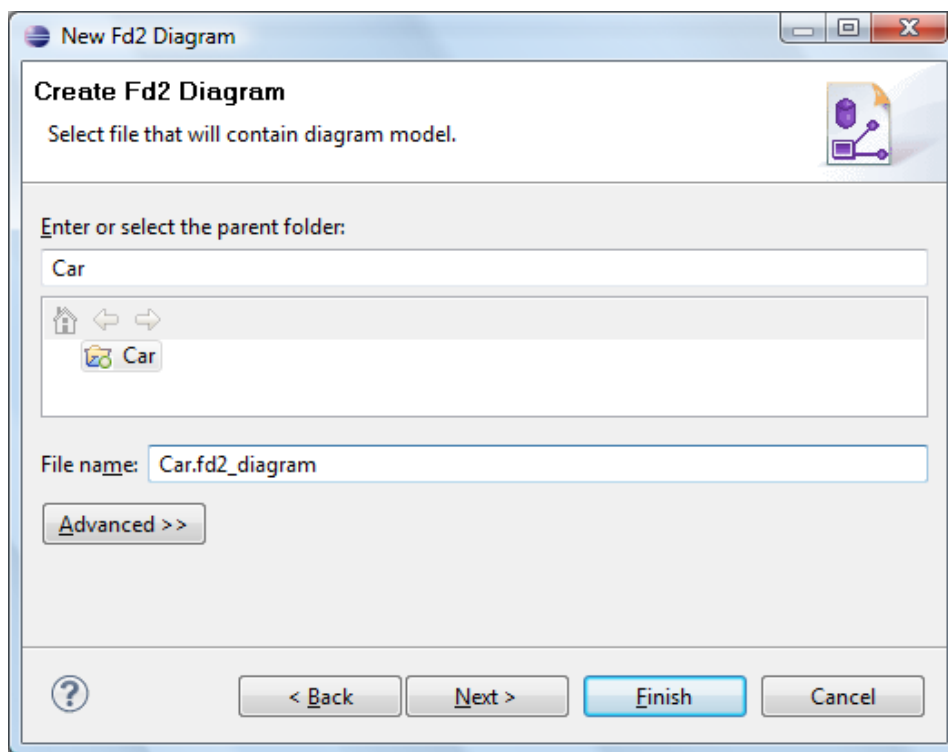
4 paveikslas. Naujos diagramos kūrimas.

Atsivėrusiame lange (žiūrėti) suraskite pasirinkimą pavadinimu „Fd2 Diagram“ arba „Fd2 Model“ (Apie skirtumus tarp Modelio ir Diagramos skaitykite skyriuje: Error: Reference source not found).



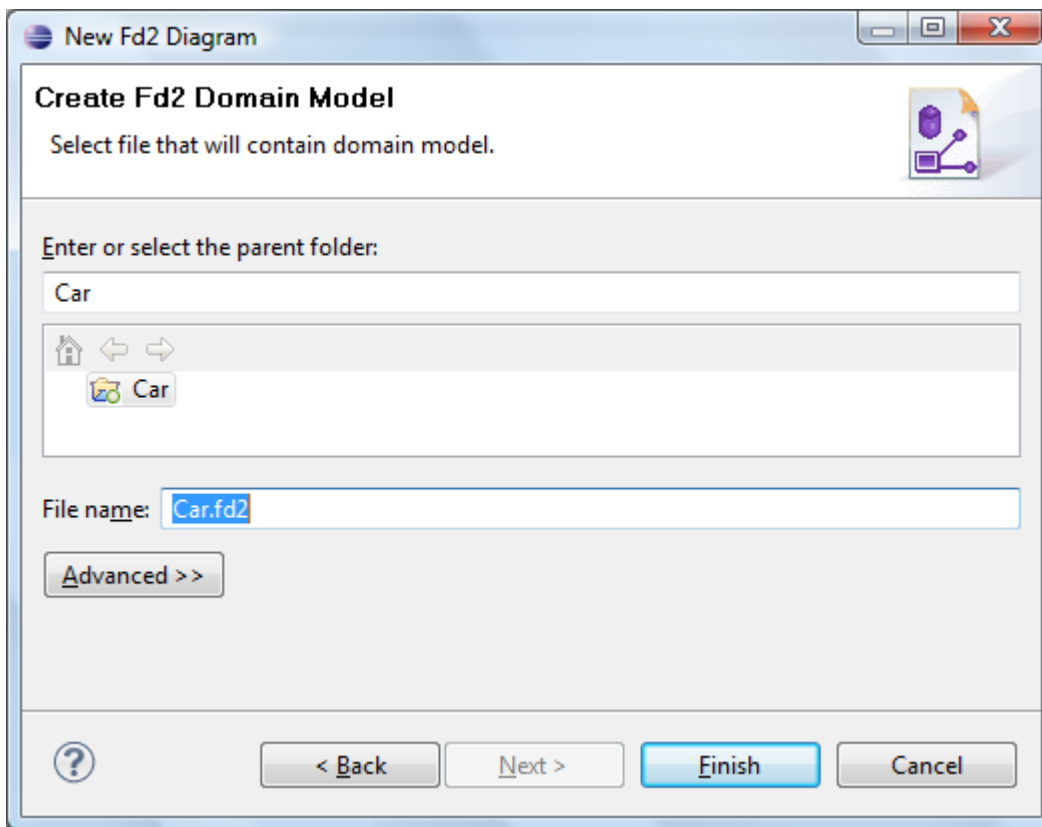
5 paveikslas. Naujo failo kūrimo dialogas.

Pasirinkus kurti Diagramos failą, programa Klaus koku pavadinimu išsaugoti diagramos failą ().  
Toliau spauskite “Finish” mygtuką.



6 paveikslas. Diagramos failo pavadinimas.

Pasirinkus kurti Požymių diagramos modelio failą, programa Klaus kokių pavadinimu išsaugoti modelio failą. Toliau spauskite “Finish” mygtuką.



7 paveikslas. Modelio failo pavadinimas.

Toliau pasirodys naujos diagramos tuščias langas, kuriame galima modeliuoti pasirinktos sistemos požymius.

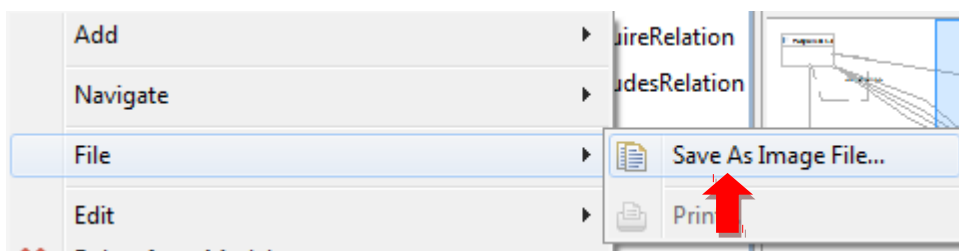
a) Galimos klaidos

Vartotojas gali nurodyti klaidingą failo pavadinimą. Tam pataisyti tereikia nurodyti tinkamą pavadinimą. Tinkamas pavadinimas formuojamas kaip: „ModelioPavadinimas.fd2“ arba „DiagramosPavadinimas.fd2\_diagram“.

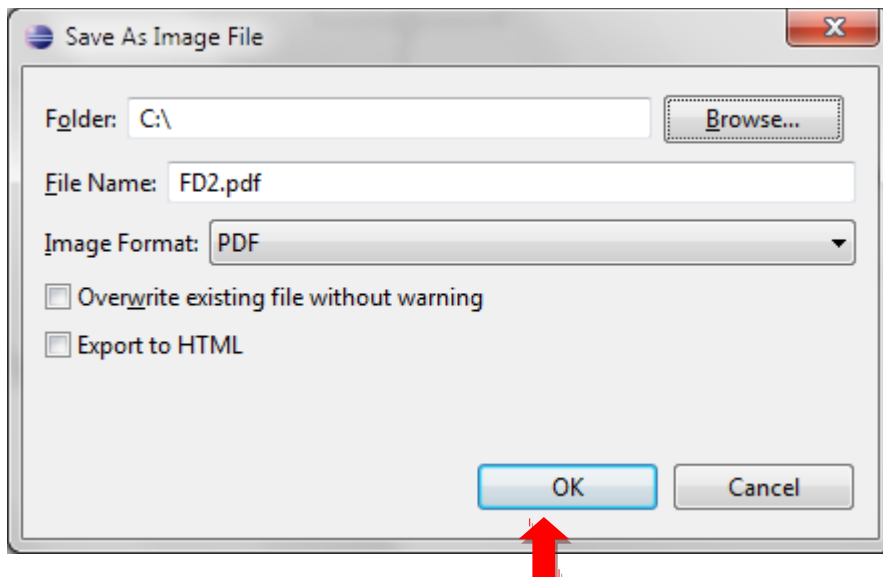
### Diagramos saugojimas į failą

Sukūrus projektą diagrama yra saugojama \*.fd2\_diagram dokumente. Norėdami užsaugoti diagramą naudokite komandas „File->Save“ arba „File->Save As...“ kad išsaugoti kaip atsarginę kopiją. Taip pat tai galima padaryti klaviatūros nuorodas (CTRL+S arba SHIFT+CTRL+S).

Norėdami diagramą išsaugoti Kitais formatais turite diagramos redagavimo lange paspausti dešinįjį klavišą ir pasirinkti „File->Save As Image File...“



8 paveikslas. Dokumento išsaugojimo kitu formatu pasirinkimas.



9 paveikslas. Diagramos išsaugojimas kitu formatu.

Atsidariusiame lange galite nurodyti kur norite išsaugoti diagramos dokumentą („File“). Galite nurodyti dokumento pavadinimą („File name“) ir pasirinkti formatą („Image Format“) iš galimų: GIF, BMP, JPEG, JPG, SVG, PNG, PDF, HTML, XML, OWL. Nustačius reikiamus parametrus spaudžiame „OK“ ir diagrama išsaugojama dokumente nurodytoje vietoje.

### Failo atidarymas

Atidaryti pasirinktą failą galima per meniu punktą „File→OpenFile“. Paspaudus šį meniu punktą pasirodo failo pasirinkimo dialogas. Jame nurodomas failas, kurį norite atidaryti.

#### c) Galimos klaidos

Sistema gali neatpažinti skaitomo failo. Galimi sprendimai:

- Patikrinkite ar atidaromas failas yra Požymių modelio failas sukurtas su šiuo



įrankiu.

- Patikrinkite ar failo galūnė yra .fd2
- Patikrinkite ar į Eclipse sistemą yra instaliuotas Požymių modeliavimo įrankis.

## **Instaliavimo vadovas**

### **Sistemos reikalavimai**

Kad sistema pasileistu ir sėkmingai veiktų reikalinga viena iš operacinių sistemų – Windows, Linux. Taip pat reikalinga įdiegta Java SE Development Kit (JDK) 5.0 Update 19 (version 1.5.0\_19) arba JDK 6 Update 14 arba naujesnė versija.

#### **7.1.1.1 Minimalūs kompiuterio reikalavimai**

*Windows operacinėms sistemoms:*

- [3] Procesoriaus: 500 MHz Intel Pentium III workstation arba panašaus;
- [4] Atminties: 256 MB RAM;
- [5] Vietos kietame diske: 500 MB.

*Linux operacinėms sistemoms:*

- [6] Procesoriaus: 500 MHz Intel Pentium III workstation arba panašaus;
- [7] Atminties: 256 MB RAM;
- [8] Vietos kietame diske: 500 MB.

#### **7.1.1.2 Rekomenduojami reikalavimai**

*Windows operacinėms sistemoms:*

- [9] Procesoriaus: 780 MHz Intel Pentium III workstation arba panašaus;
- [10] Atminties: 512 MB RAM;
- [11] Vietos kietame diske: 800 MB.

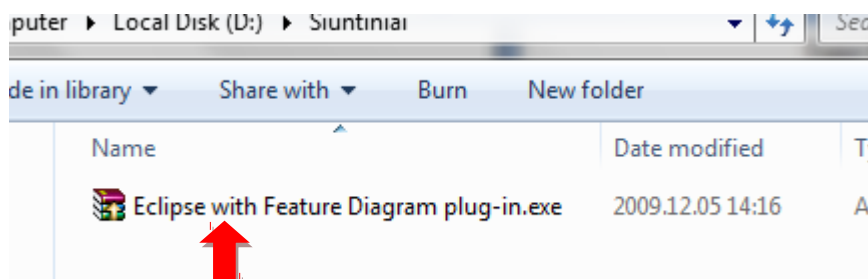
*Linux operacinėms sistemoms:*

- [12] Procesoriaus: 780 MHz Intel Pentium III workstation arba panašaus;
- [13] Atminties: 512 MB RAM;
- [14] Vietos kietame diske: 800 MB.

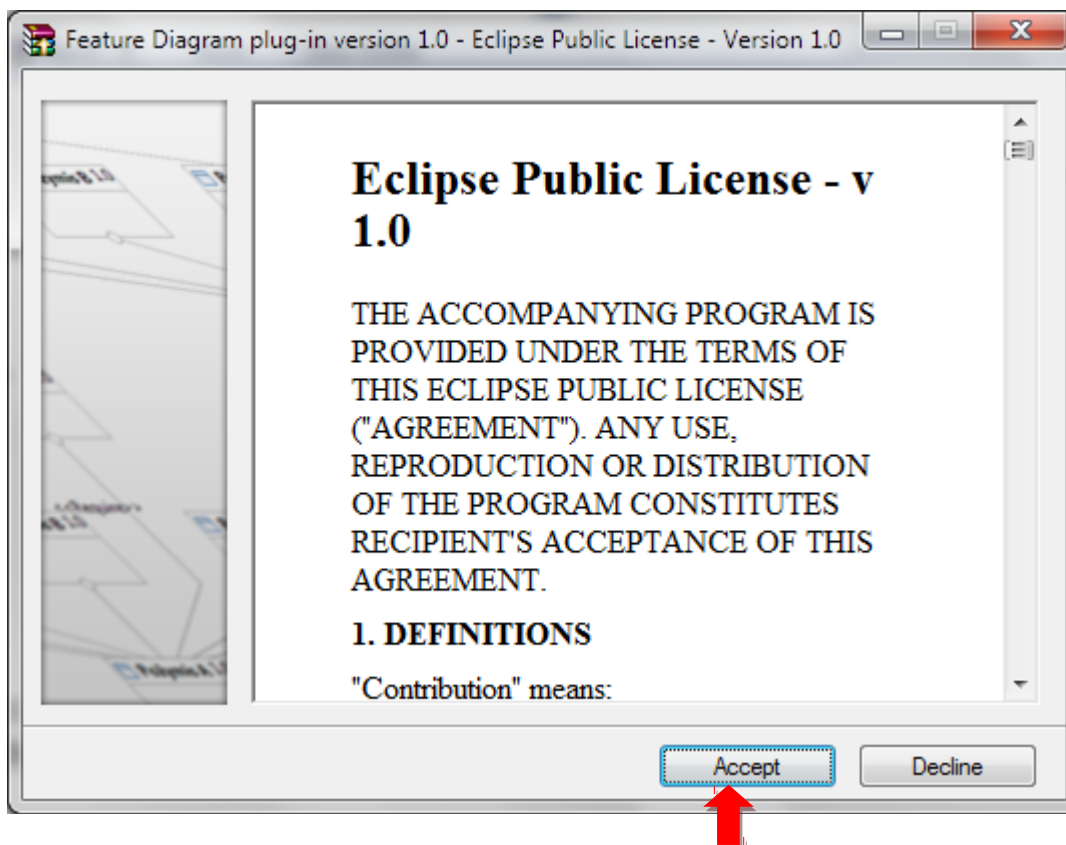
## Diegimo žingsniai

### 7.1.1.3 Windows operacinėje sistemoje

Paleidus programos instaliacinį dokumentą atsidaro licencijos dokumentas, kuris prašo sutikti arba nesutikti su licencijos taisyklėmis.



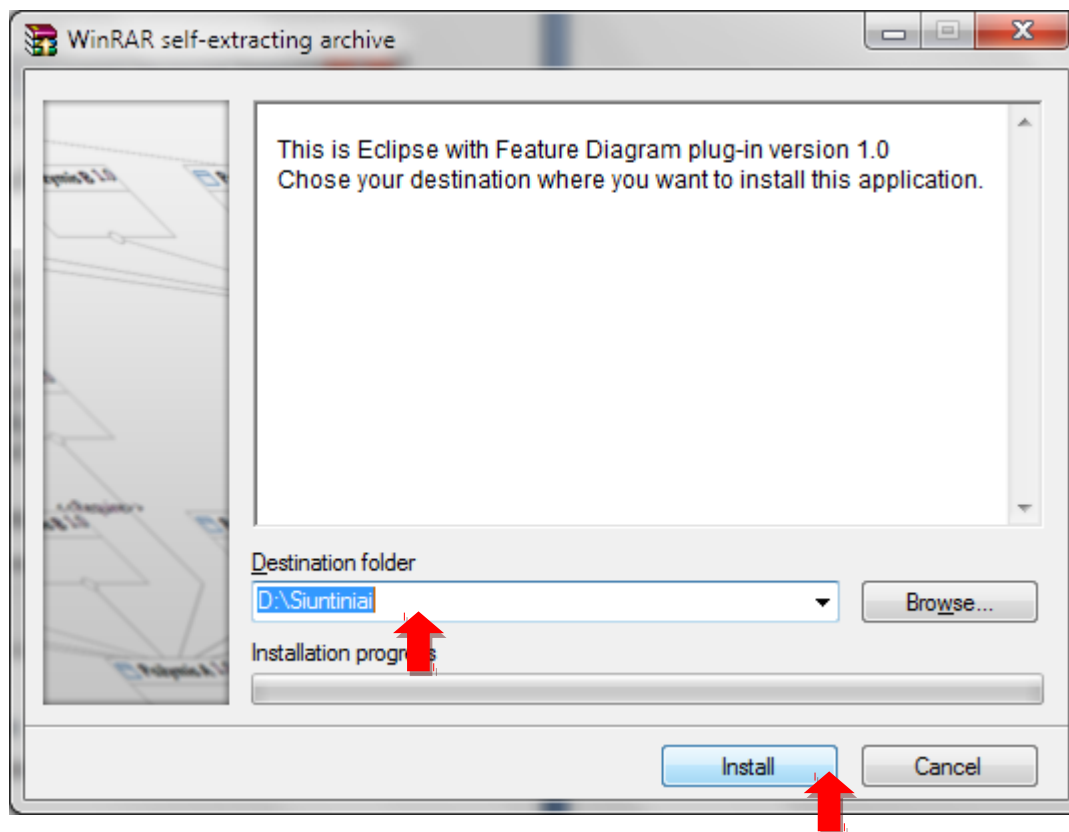
10 paveikslas. Programos instaliacinis dokumentas.



11 paveikslas. Licencijos langas.

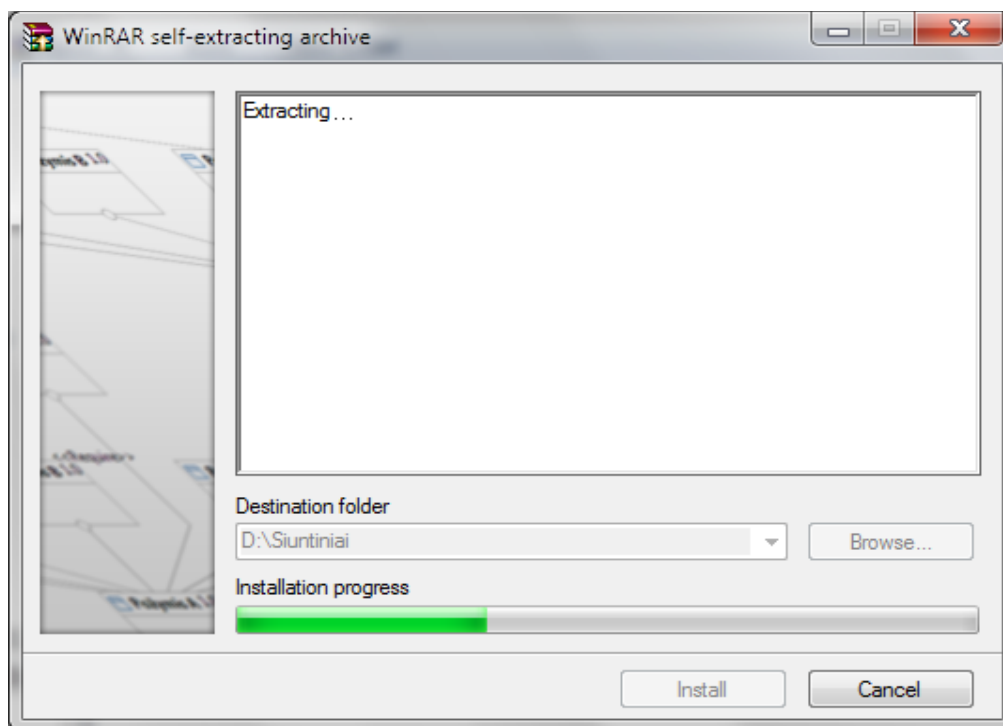
Jeigu norite naudotis programa, Jūs turite sutikti su licencijos taisyklėmis ir paspausti „Accept“ mygtuką, jeigu nesutinkate, spauskite „Decline“, tačiau nesutikus su licencijos taisyklėmis programos diegimas bus baigtas

Sutikus su taisyklėmis atidaromas diegimo nustatymų langas.



12 paveikslas. Diegimo nustatymų langas.

Diegimo nustatymų lange jums reikia nurodyti vietą kompiuteryje kur norėtumėte įdiegti naują programą (7.1.1.3). Tai padaryti galima paspaudus mygtuką „Browse...“. Nustačius vietą pradedame programos diegimą.



13 paveikslas. Diegimo nustatymų langas (diegimo procesas).

Pasibaigus programos diegimui langas uždaromas ir pasileidžia požymių diagramų modeliavimo įrankis.

#### **7.1.1.4 Linux operacinėse sistemose**

Linux operacinėse sistemose diegimas yra paprastesnis. Sistema pateikiama dviem dokumentais - instaliaciniame dokumente ir archyvuotame Zip dokumente. Linux operacinėse sistemose naudojamas Zip archyvuotas dokumentas. Užtenka išarchyvuoti archyvo turinį į norimą kompiuterio vietą ir paleisti paleidimo failą „eclipse.exe“.

## **8 PAKEITIMŲ SĄRAŠAS**

Pakeitimų nebuvo

## **9 LITERATŪRA**

[1] AHEAD architektūra <http://www.cs.utexas.edu/users/schwartz/ATS.html>

- [2] XFeature <http://www.pnp-software.com/XFeature/Home.html>
- [3] Feature modeling plugin <http://gsd.uwaterloo.ca/projects/fmp-plugin/>
- [4] Ecore Feature modeling plugin <http://gsd.uwaterloo.ca/projects/ecorefmp/>
- [5] CaptainFeature <http://sourceforge.net/projects/captainfeature/>
- [6] RequiLine\_ <http://www-lufgi3.informatik.rwth-aachen.de/TOOLS/requiline/index.php>
- [7] FAMA <http://www.isa.us.es/fama/>
- [8] Gears, <http://www.biglever.com/>
- [9] Kumbang Tools, <http://www.soberit.hut.fi/KumbangTools>
- [10] K. Kang, S. Cohen, J. Hess, W. Novak, and S. Peteson. Feature-Oriented Domain Analysis (FODA) Feasibility Study. Technical Report CMU/SEI-90-TR-21, Software Engineering Institute, Carnegie Mellon University, Nov. 1990.
- [11] R. Damaševičius, 2002. Specification of Learning Content using Feature Diagrams.
- [12] V.Štuikys, R. Damaševičius. Exploration of Learning Object Ontologies Using Feature Diagrams.
- [13] XML informacinis puslapis [<http://en.wikipedia.org/wiki/XML>]
- [14] XMI informacinis puslapis [[http://en.wikipedia.org/wiki/XML\\_Metadata\\_Interchange](http://en.wikipedia.org/wiki/XML_Metadata_Interchange)]
- [15] OWL oficialus puslapis [<http://www.w3.org/TR/2004/REC-owl-guide-20040210/#TermIndex>]
- [16] OWL informacinis puslapis [[http://en.wikipedia.org/wiki/Web\\_Ontology\\_Language](http://en.wikipedia.org/wiki/Web_Ontology_Language)]
- [17] OWL klasių aprašas [<http://www.w3.org/TR/2004/REC-owl-guide-20040210/#ComplexClasses>]
- [18] EBNF informacinis puslapis [<http://en.wikipedia.org/wiki/EBNF>]
- [19] EBNF gramatikos analizė [[http://www.boost.org/community/feature\\_model\\_diagrams.html](http://www.boost.org/community/feature_model_diagrams.html)]
- [20] Požymių diagramų informacinis puslapis [[http://en.wikipedia.org/wiki/Feature\\_model](http://en.wikipedia.org/wiki/Feature_model)]

## 10 TERMINŲ IR SANTRUMPŲ ŽODYNAS

1. **OMG** (*Object Management Group*) — konsorciumas (darbinė grupė), užsiimanti objektinių technologijų ir standartų vystymu
2. **MOF** (*Meta Object Facility*) - OMG meta-modeliavimo ir meta-duomenų saugyklos standartas

3. Eclipse įskiepis: I Eclipse IDE integruotas įrankis kuris atlieka tam tikrą funkciją
4. Eclipse (platforma) – programa, priedų platforma, kurią galima papildyti individualiais priedais