**UNIVERZITET U KRAGUJEVCU**

**FAKULTET TEHNIČKIH NAUKA ČAČAK**



**SEMINARSKI RAD**

Predmet: **Inteligentni sistemi**

TEMA: **Ekspetni sistem u Drools programu**

Primer: **Dijagnoza kvarova mobilnih telefona**

Profesor: Student:

**dr Danijela Milošević Igor Dimitrijević 139/2012**

Asistent:

**mr Vanja Luković**

**Čačak, 2015.**

**Sadržaj:**

[Drools 1](#_Toc3320)

[Drools Expert - osnovne karakteristike 1](#_Toc10181)

[Drools Expert - sintaksa 1](#_Toc6754)

[Instalacija Drools - a 2](#_Toc10476)

[Kreiranje Drools projekta 3](#_Toc31708)

[DroolsTest.java 3](#_Toc12882)

[Mobilni.java 5](#_Toc16463)

[KvaroviMobilnihTelefona.drl 7](#_Toc1198)

[Mobilni se uključuje, ali se restartuje 7](#_Toc9532)

[Mobilni se ukljucuje i ne prepoznaje SD karticu 9](#_Toc4221)

[Mobilni se ukljucuje, ali ne reprodukuje zvuk 10](#_Toc26157)

[Mobilni se ukljucuje, ali ekran ne prikazuje sadrzaj 12](#_Toc25652)

[Mobilni se ne puni 13](#_Toc23689)

[Ekran ne reaguje na dodir 14](#_Toc3395)

[Mobilni se ne ukljucuje 15](#_Toc6936)

[Zaključak 16](#_Toc2558)

[Literatura 17](#_Toc19145)

# Drools

Drools je napisan u Javi i besplatan je. Koristi pravila i Java objekte za predstavljanje znanja i ulančavanje unapred.

Strategije za rešavanje konflikta:

* prioritet pravila (salience)
* svako pravilo se izvršava samo jednom (no-loop, lock-on-active).

## Drools Expert - osnovne karakteristike

Baza znanja se čuva u DRL formatu (format karakterističan samo za ovaj shell). Alati za razvoj i testiranje su dati kao Eclipse plug-in. Dat je u formi framework-a (skupa gotovih klasa). Činjenice se unose u vidu Java objekata. Veoma lako se integriše u bilo koju Java aplikaciju (direktan pristup Java klasama).

## Drools Expert - sintaksa

Pravila se nalaze u DRL fajlovima (DRL - Drools Rule Language). Jedan fajl može da sadrži više pravila, a ES se može sastojati iz više DRL fajlova.

Struktura DRL fajla:

*package ime\_paketa\_za\_pravila;*

*import java\_klasa\_1;*

*import java\_klasa\_2;*

*...*

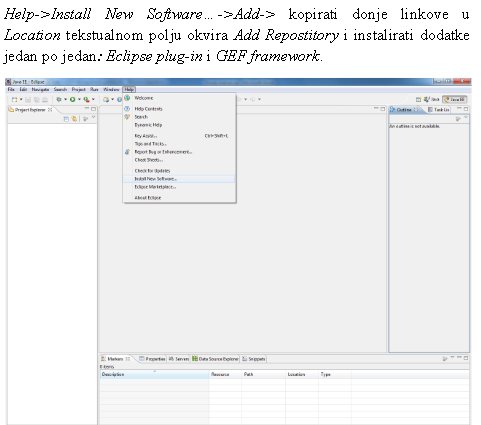
*//pravila*

# Instalacija Drools - a

Da biste instalirali Drools potrebno je da imate vec instaliran Eclipse. Nakon toga preuzmete Drools sa sledeceg sajta <http://www.jboss.org/drools/downloads>.

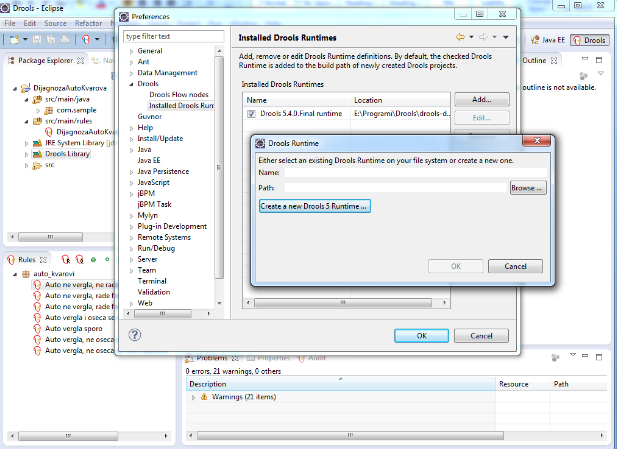
Potrebno je instalirati i dodatke u Eclipse program (slika 1):

* Eclipse plug-in se skida sa sajta: <http://download.jboss.org/drools/release/5.4.0.Final/org.drools.updatesite/>
* Eclipse GEF framework se skida sa sajta: [http://download.eclipse.org/tools/gef/updates/releases/](http:/download.eclipse.org/tools/gef/updates/releases/)



*Slika 1 - Instaliranje potrebnih Drools dodataka*

Podesiti Runtime Drools okruženje: Window/Preferences/Drools/Installed Drools Runtime environments/Create a new Drools 5 Runtime/definisati putanju do binaries/ direktorijuma Drools programa skinutog sa sajta. Nakon toga resetovati Eclipse program (slika 2).



*Slika 2 - Runtime Drools okruženje*

# Kreiranje Drools projekta

Kreirati Drools projekat u okviru Eclipse alata. Nakon kreiranja novog Drools projekta, uočavaju se dve fascikle: *src/main/java* i *src/main/rules*. Fascikla *src/main/java* sadrži *com/sample* fasciklu u kojoj se nalazi *DroolsTest.java* fajl i klasa koja se u slučaju koji sam ja radio zove *Mobilni.java* a facikla *src/main/rules* sadrži *KvaroviMobilnihTelefona.drl* fajl.

*DroolsTest.java* fajl sadrži glavnu funkciju Java programa (*main()*), u kojoj je formirana baza znanja i u koju su učitana pravila iz *KvaroviMobilnihTelefona.drl* fajla (funkcija *readRules()*), definisane činjenice, ubačene u radnu memoriju i nakon toga je potrebno pokrenuti izvršavanje pravila.

## DroolsTest.java

package com.sample;

import java.awt.Desktop;

import java.io.InputStreamReader;

import java.io.Reader;

import org.drools.RuleBase;

import org.drools.RuleBaseFactory;

import org.drools.WorkingMemory;

import org.drools.compiler.PackageBuilder;

import org.drools.rule.Package;

import org.goodoldai.jeff.wizard.JEFFWizard;

public class DroolsTest {

public static final void main(String[] args) {

try {

//Ucitavanje baze znanja

RuleBase ruleBase = readRules();

//Kreiranje radne memorije

WorkingMemory workingMemory = ruleBase.newStatefulSession();

//Kreiranje instance mehanizma za objasnjavanje

JEFFWizard ef = new JEFFWizard();

ef.setInternationalization(false);

//Postavljanje naslova objasnjenja i kreiranje praznog objasnjenja

//Kreiranje cinjenica

Mobilni mob = new Mobilni();

mob.setUkljucujeSe("Da");

mob.setRestartujeSe("Ne");

mob.setDetektujeSDkarticu("Ne");

//Pravila

//Unosenje cinjenica u radnu memoriju

workingMemory.insert(mob);

//Unosenje instance mehanizma za objasnjavanje u radnu memoriju kao globalne promenljive

workingMemory.setGlobal("ef", ef);

//Pokretanje ES

workingMemory.fireAllRules();

//Ispisivanje rezultata

System.out.println(mob);

//Pravljenje PDF fajla koji sadrzi objasnjenje i njegovo prikazivanje

ef.generatePDFReport("Objasnjenje.pdf",false);

if (Desktop.isDesktopSupported())

java.awt.Desktop.getDesktop().browse(java.net.URI.create((new java.io.File("Objasnjenje.pdf").toURI().toString())));

} catch (Throwable t) {

t.printStackTrace();

}

}

private static RuleBase readRules() throws Exception {

//Ucitavanje pravila iz fajla

Reader source = new InputStreamReader( DroolsTest.class.getResourceAsStream( "/KvaroviMobilnihTelefona.drl" ) );

//Kreiranje paketa koji sadrzi sva pravila

PackageBuilder builder = new PackageBuilder();

builder.addPackageFromDrl( source );

Package pkg = builder.getPackage();

//Kreiranje baze znanja i dodavanje paketa u nju

RuleBase ruleBase = RuleBaseFactory.newRuleBase();

ruleBase.addPackage( pkg );

return ruleBase;

}

}

## Mobilni.java

Definisana klasa *Mobilni.java* samo sadrži privatne attribute, javne *Get* i *Set* metode za pristup i *ToString()* metodu i konstante u kojima se čuvaju putanje do slika. Naredne linije koda predstavljaju gore pomenutu klasu:

**package** com.sample;

**public** **class** Mobilni {

// konstante - imena slika

**public** **static** **final** String *bootloop* = "/bootloop.jpg";

**public** **static** **final** String *sd* = "/sd-slot.jpg";

**public** **static** **final** String *speaker* = "/speaker.jpg";

**public** **static** **final** String *flatCable* = "/flat-cable.jpg";

**public** **static** **final** String *charging* = "/charging.jpg";

**public** **static** **final** String *digitizer* = "/digitizer.jpg";

**public** **static** **final** String *battery* = "/battery.jpg";

// polja

**private** String ukljucujeSe = "nepoznato";

**private** String restartujeSe = "nepoznato";

**private** String reprodukujeZvuk = "nepoznato";

**private** String radiEkran = "nepoznato";

**private** String puniSe = "nepoznato";

**private** String detektujeSDkarticu = "nepoznato";

**private** String reagujeNaDodir = "nepoznato";

**private** String uzrokProblema = "nepoznato";

**private** String resenje = "nepoznato";

**public** String getUkljucujeSe() {

**return** ukljucujeSe;

}

**public** **void** setUkljucujeSe(String ukljucujeSe) {

**this**.ukljucujeSe = ukljucujeSe;

}

**public** String getRestartujeSe() {

**return** restartujeSe;

}

**public** **void** setRestartujeSe(String restartujeSe) {

**this**.restartujeSe = restartujeSe;

}

**public** String getReprodukujeZvuk() {

**return** reprodukujeZvuk;

}

**public** **void** setReprodukujeZvuk(String reprodukujeZvuk) {

**this**.reprodukujeZvuk = reprodukujeZvuk;

}

**public** String getRadiEkran() {

**return** radiEkran;

}

**public** **void** setRadiEkran(String radiEkran) {

**this**.radiEkran = radiEkran;

}

**public** String getPuniSe() {

**return** puniSe;

}

**public** **void** setPuniSe(String puniSe) {

**this**.puniSe = puniSe;

}

**public** String getDetektujeSDkarticu() {

**return** detektujeSDkarticu;

}

**public** **void** setDetektujeSDkarticu(String detektujeSDkarticu) {

**this**.detektujeSDkarticu = detektujeSDkarticu;

}

**public** String getReagujeNaDodir() {

**return** reagujeNaDodir;

}

**public** **void** setReagujeNaDodir(String reagujeNaDodir) {

**this**.reagujeNaDodir = reagujeNaDodir;

}

**public** String getUzrokProblema() {

**return** uzrokProblema;

}

**public** **void** setUzrokProblema(String uzrokProblema) {

**this**.uzrokProblema = uzrokProblema;

}

**public** String getResenje() {

**return** resenje;

}

**public** **void** setResenje(String resenje) {

**this**.resenje = resenje;

}

**public** String toString(){

**return** "Uzrok problema: "+uzrokProblema+"\nResenje:\n"+resenje;

}

}

## KvaroviMobilnihTelefona.drl

Ono na čemu ceo sistem počiva jesu pravila koja predstavljaju bazu znanja a nalaze se u *KvaroviMobilnihTelefona.drl.* Najpre je potrebno kreirati pravilo, zatim utvrditi koji su to problemi koji izazivaju kvar, tj. koji su uslovi potrebni da budu ispunjeni da bi pravilo bilo pokrenuto. U tekstu koji sledi biće predočeni kvarovi, kao i rešenja tj. način rešavanja problema datih kvarova.

### Mobilni se uključuje, ali se restartuje

***Kod pravila:***

**rule** "Mobilni se ukljucuje, ali se restartuje"

**lock-on-active** **true**

**when**

mob : Mobilni(ukljucujeSe == "Da", restartujeSe == "Da")

**then**

mob.setUzrokProblema("Softverski kvar");

mob.setResenje("1. Mobilni uredjaj odneti kod ovlascenog servisera");

**update**(mob);

ef.setTitle("Softverski kvar - bootloop");

ef.createExplanation();

ef.addText(**null**, "Ova greska je jedna od najcescih gresaka softverske prirode, tzv. bootloop, tj. stanje kada se mobilni telefon neprekidno restartuje je jasan indikator da postoji odredjeni problem sa softverom telefona, tacnije sa operativnim sistemom."+

"\nMobilni uredjaj je potrebno sto pre odneti kod ovlascenog servisera koji ce utvrditi uzrok otkazivanja softvera.");

ef.addImage(**new** ImageData(Mobilni.bootloop));

ef.addText(**null**, "Bootloop mobilnog telefona");

**end**

**Postupak rešavanja određenog problema (Mobilni se uključuje, ali se restartuje):**



***Rešenje detaljnije opisano u pdf fajlu:***



### Mobilni se ukljucuje i ne prepoznaje SD karticu

***Kod pravila:***

**rrule** "Mobilni se ukljucuje i ne prepoznaje SD karticu"

**lock-on-active** **true**

**when**

mob : Mobilni(ukljucujeSe == "Da", restartujeSe == "Ne", detektujeSDkarticu == "Ne")

**then**

mob.setUzrokProblema("Ne detektuje SD karticu");

mob.setResenje(

"1. Mozete pokusati sami da otklonite kvar\n"+

" 1.1. Skinite zadnji poklopac telefona\n"+

" 1.2. Pronadjite slot za SD karticu\n"+

" 1.3. Izvadite SD karticu\n"+

" 1.4. Produvajte SD slot\n"+

" 1.5. Vratite SD karticu u slot i restartujte telefon\n\n ili,\n\n"+

"2. Mobilni uredjaj odneti kod ovlascenog servisera"

);

**update**(mob);

ef.setTitle("Kvar SD slota");

ef.createExplanation();

ef.addText(**null**,

"Ovaj takozvani kvar je moguce lako otkloniti ukoliko se radi o prasini koja sprecava pravilno funkcionisanje istog.\n"+

"Ukoliko ni nakon primene datog resenja telefon ne detektuje SD karticu, potrebno je uredjaj odneti kod ovlascenog servisera."

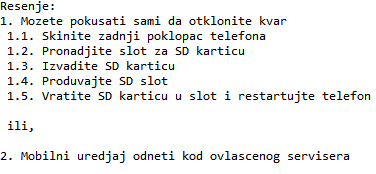
);

ef.addImage(**new** ImageData(Mobilni.sd));

ef.addText(**null**, "SD slot");

**end**

***Postupak rešavanja određenog problema (Mobilni se ukljucuje i ne prepoznaje SD karticu):***



***Rešenje detaljnije opisano u pdf fajlu:***



### Mobilni se ukljucuje, ali ne reprodukuje zvuk

***Kod pravila:***

**rule** "Mobilni se ukljucuje, ali ne reprodukuje zvuk"

**lock-on-active** **true**

**when**

mob : Mobilni(ukljucujeSe == "Da", reprodukujeZvuk == "Ne")

**then**

mob.setUzrokProblema("Ne reprodukuje zvuk");

mob.setResenje(

"1. Proverite da li postoji fizicko ostecenje zvucnika\n"+

"1.1. Proverite da nije samo smanjena jacina zvuka\n"+

"1.2. Proverite da nije upala prasina u zvucnik\n"+

"2. Mobilni uredjaj odneti kod ovlascenog servisera"

);

**update**(mob);

ef.setTitle("Fizicki kvar - zvucnik");

ef.createExplanation();

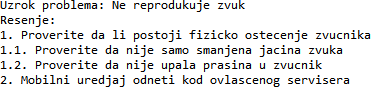
ef.addText(**null**, "Ukoliko nakon predlozenih koraka zvucnik i dalje ne reprodukuje zvuk, telefon odnesite kod ovlascenog servisera.");

ef.addImage(**new** ImageData(Mobilni.speaker));

ef.addText(**null**, "Zvucnik");

**end**

***Postupak rešavanja određenog problema (Mobilni se ukljucuje, ali ne reprodukuje zvuk):***



***Rešenje detaljnije opisano u pdf fajlu:***



### Mobilni se ukljucuje, ali ekran ne prikazuje sadrzaj

***Kod pravila:***

**rule** "Mobilni se ukljucuje, ali ekran ne prikazuje sadrzaj"

**lock-on-active** **true**

**when**

mob : Mobilni(ukljucujeSe == "Da", radiEkran == "Ne")

**then**

mob.setUzrokProblema("Ekran ne prikazuje sadrzaj");

mob.setResenje("1. Uredjaj odneti kod ovlascenog servisera, posto je potrebno zameniti flet kabl ekrana.");

**update**(mob);

ef.setTitle("Fizicki kvar - ekran");

ef.createExplanation();

ef.addText(**null**, "Uredjaj odneti kod ovlascenog servisera, posto je potrebno zameniti flet kabl ekrana.");

ef.addImage(**new** ImageData(Mobilni.flatCable));

ef.addText(**null**, "Flet kabl");

**end**

***Postupak rešavanja određenog problema (Mobilni se ukljucuje, ali ekran ne prikazuje sadrzaj):***



***Rešenje detaljnije opisano u pdf fajlu:***



### Mobilni se ne puni

***Kod pravila:***

**rule** "Mobilni se ne puni"

**lock-on-active** **true**

**when**

mob : Mobilni(puniSe == "Ne")

**then**

mob.setUzrokProblema("Mobilni telefon se ne puni");

mob.setResenje("1. Uredjaj odneti kod ovlascenog servisera, posto je potrebno zameniti konektor za punjenje.");

**update**(mob);

ef.setTitle("Fizicki kvar - konektor za punjenje");

ef.createExplanation();

ef.addText(**null**, "Uredjaj odneti kod ovlascenog servisera, posto je potrebno zameniti konektor za punjenje.");

ef.addImage(**new** ImageData(Mobilni.charging));

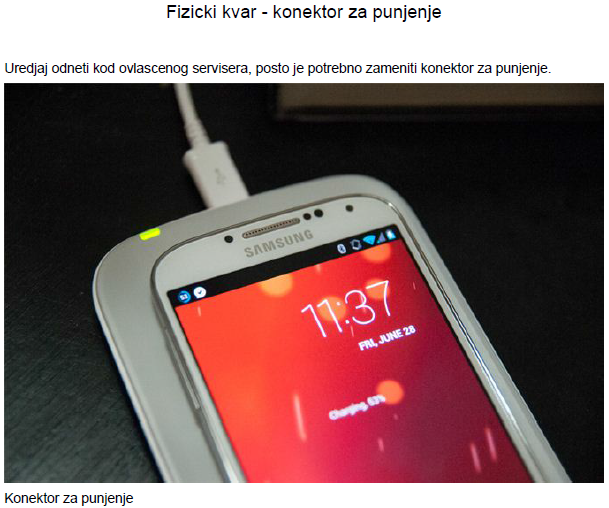
ef.addText(**null**, "Konektor za punjenje");

**end**

***Postupak rešavanja određenog problema (Mobilni se ne puni):***



***Rešenje detaljnije opisano u pdf fajlu:***



### Ekran ne reaguje na dodir

***Kod pravila***

**rule** "Ekran ne reaguje na dodir"

**lock-on-active** **true**

**when**

mob : Mobilni(ukljucujeSe == "Da", radiEkran == "Da", reagujeNaDodir == "Ne")

**then**

mob.setUzrokProblema("Digitajzer");

mob.setResenje(

"1. Restartovati telefon, ili koriscenjem fizickog dugmeta ili vadjenjem baterije\n"+

"2. Uredjaj odneti kod ovlascenog servisera, posto je potrebno zameniti digitajzer ekrana"

);

**update**(mob);

ef.setTitle("Fizicki kvar - digitajzer");

ef.createExplanation();

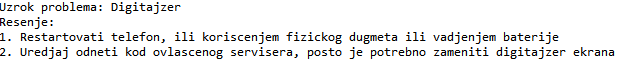
ef.addText(**null**, "Uredjaj odneti kod ovlascenog servisera, posto je potrebno zameniti digitajzer ekrana.");

ef.addImage(**new** ImageData(Mobilni.digitizer));

ef.addText(**null**, "Digitajzer");

**end**

***Postupak rešavanja određenog problema (Ekran ne reaguje na dodir):***



***Rešenje detaljnije opisano u pdf fajlu:***



### Mobilni se ne ukljucuje

***Kod pravila***

**rule** "Mobilni se ne ukljucuje"

**lock-on-active** **true**

**when**

mob : Mobilni(ukljucujeSe == "Ne")

**then**

mob.setUzrokProblema("Prazna baterija");

mob.setResenje(

"1. Telefon povezati na punjac\n\n"+

"Ukoliko se telefon nije ukljucio odneti kod ovlascenog servisera."

);

**update**(mob);

ef.setTitle("Prazna baterija");

ef.createExplanation();

ef.addText(**null**,

"Najcesci uzrok ovog problema je baterija koja je skroz ispraznjena. Telefon povezati na punjac i ostaviti par minuta da se napuni.\n\n"+

"Ukoliko se telefon nije ukljucio odneti kod ovlascenog servisera."

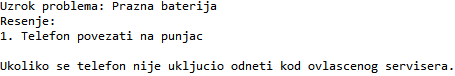
);

ef.addImage(**new** ImageData(Mobilni.battery));

ef.addText(**null**, "Baterija");

**end**

***Postupak rešavanja određenog problema (Mobilni se ne ukljucuje):***



***Rešenje detaljnije opisano u pdf fajlu:***



# Zaključak

Cilj ovog rada bio je pravljenje ekspertnog sistema koji će na osnovu opisa određenog kvara predložiti odgovarajuće popravke, što je i prikazano u gore navedenom kodu, koji je podržan slikama za odgovarajuće kvarove kao i njihova rešenja.

# Literatura

1. [www.google.com](http://www.google.com/)
2. <http://itlab.ftn.kg.ac.rs/moodle>