

## Poglavje 3

### DIA - Načrtovanje diagramov

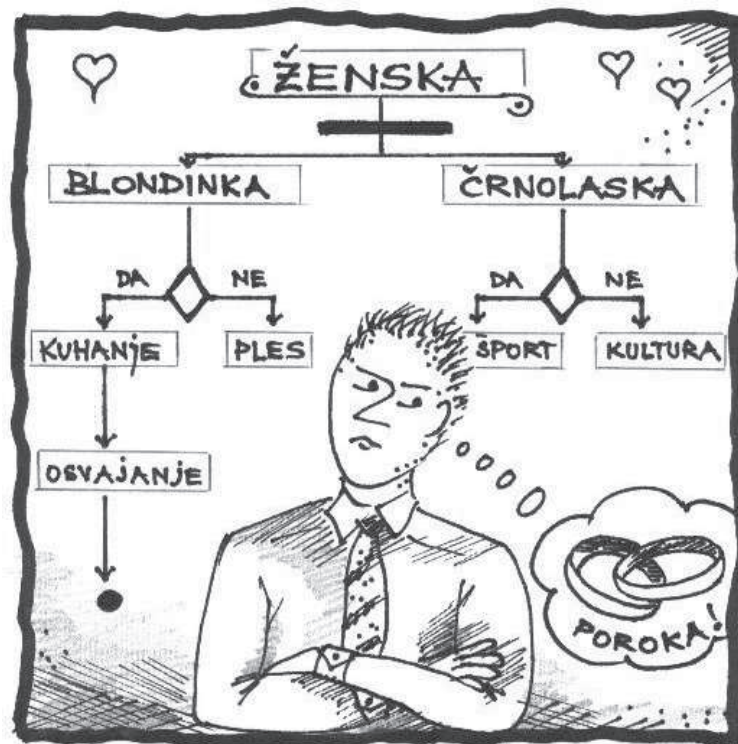


Diagram poteka (aktivnosti)  
UML jezik  
Primer: načrtovanje diagrama z UML

## 3.1 Teoretično ozadje

### 3.1.1 Diagram poteka/aktivnosti

Diagram poteka (*Flow chart*) je diagram za prikaz možnih poti podatkov skozi sistem oz. je eden izmed načinov zapisa algoritma. Diagram prikazuje natančno zaporedje operacij, ki jih programsko orodje pri obdelavi podatkov izvede. Različni grafični simboli predstavljajo vnos in izpis podatkov, odločitve, razvejitve in podprogramska orodja. Uporablja se v računalništvu, matematiki, pravu, logistiki in v mnogih drugih vedah. Danes se diagrami poteka oz. diagrami aktivnosti rišejo predvsem v standardiziranem opisnem jeziku UML (*Unified Modeling Language*).

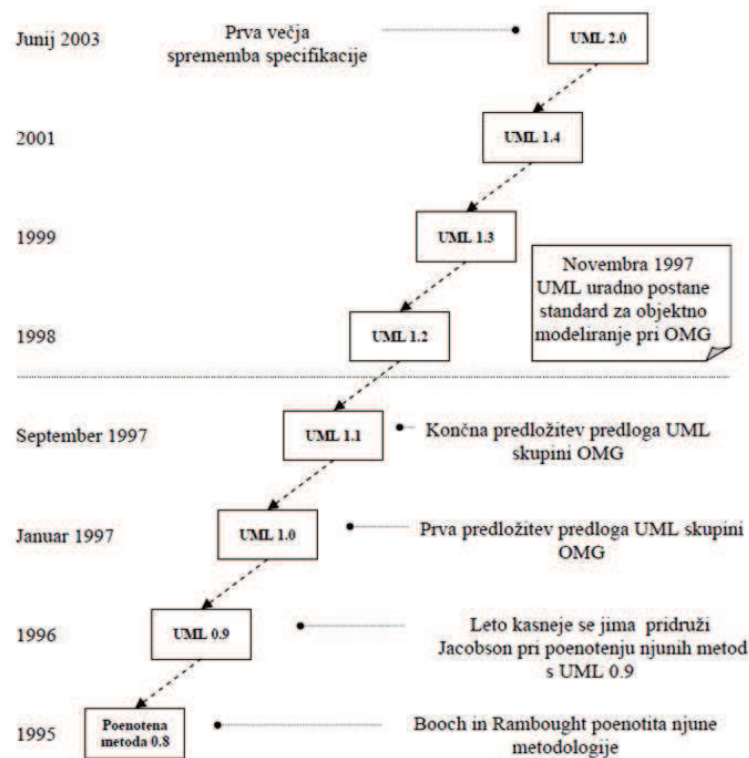
Diagram poteka prikazuje prehode iz ene aktivnosti na drugo. Elementi diagrama aktivnosti so: aktivnosti, prehodi med aktivnostmi, razvejitve (sočasne razvejitve), sočasno stičišče, linija toka dogodkov in objektni tok. Kot rezultat aktivnosti je akcija, katere posledica je prehod v drugo aktivnost ali pa le vračanje vrednosti. Iz diagrama je jasno razvidna odgovornost za posamezne aktivnosti. Če so v tok dogodkov vpleteni pomembni objekti, jih lahko s pomočjo objektnega toka dodamo v diagram aktivnosti.

### 3.1.2 Standard UML

Programsko orodje UML je grafični jezik za vizualizacijo, modeliranje, specifikacijo, konstruiranje in dokumentiranje programske opreme. UML predstavlja zbir najboljših postopkov v praktično objektno orientiranem modeliranju. Vedeti moramo, da UML ni metodologija za razvoj informacijskega sistema, ampak jezik, skupek tehnik, ki jih lahko uporabljamo skupaj s poenotenim procesom ali katero drugo metodologijo.

Začetek razvoja UML umeščamo v leto 1994. Takrat je bilo na voljo okoli 50 različnih metodologij za objektno usmerjeno analizo in načrtovanje. Ideja raziskovalcev, ki so raziskovali na tem področju je bila združitev metodologij v enoten jezik za modeliranje, ki bi vseboval prednosti različnih metodologij. Tako je nastal UML. Med novimi metodami, ki so se pojavile v tistem času so izstopale Boochova metoda, Jacobsonova metoda OOSE (*Object-Oriented Software Engineering*) in Rumbaughtova metoda OMT (*Object Modeling Techniq*). V sredini 90-tih let so avtorji teh treh metod zasnovali nov modelni jezik, ki naj bi bil podprt v vseh treh metodah (glej Sliko 3.1).

Cilj UML je uporabniku ponuditi standarden, izrazno močan vizualni modelirni jezik; omogočiti mehanizme razširjanja in specializacije osnovnih konceptov; biti neodvisen od določenega programskega jezika in razvojnega procesa; ponuditi formalno osnovo za razumevanje jezika za modeliranje;



Slika 3.1: Zgodovinski razvoj specifikacije poenotene modelnega jezika UML

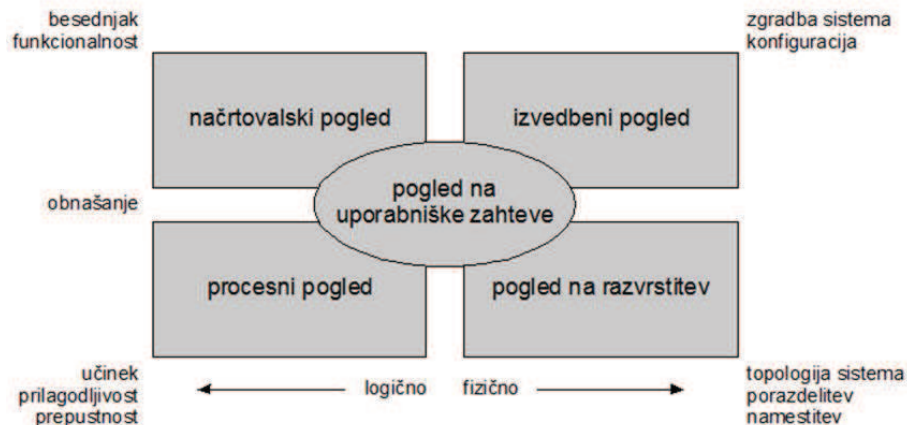
Vir: [78]

spodbuditi rast tržišča objektnih orodij; podpirati visoko nivojske razvojne koncepte, kot so sodelovanje, ogrodja, vzorci in komponente ter integrirati najboljše prakse.

Na sistem UML je moč gledati skozi pet različnih pogledov, kar prikazuje Slika 3.2.

### Modeliranje z UML

Pri modeliranju z UML, vrstni red ni predpisan. Koristno je, da najprej modeliramo pogled na uporabniške zahteve, nato načrtovalski in procesni pogled, na koncu pa še pogled na komponente in razvrstitev. Da lahko razumemo delovanje UML, je potrebno poznati pomen treh osnovnih gradnikov, in sicer: elementov, povezav in diagramov.



Slika 3.2: Pet pogledov na sistem UML

Vir: [101]

## Elementi

V UML-ju se pojavljajo štiri vrste elementov: strukturni elementi, elementi obnašanja, elementi združevanja in elementi opomb. Strukturni elementi so največkrat statični deli modela, ki predstavljajo element shematično ali fizično. Mednje prištevamo: razrede, vmesnike, sodelovanje, primere uporabe komponente in vozlišča [100].

- Razred je opis nabora objektov, ki delijo enake attribute, operacije, povezave in semantiko. Vmesnik definira povezavo med specifikacijo, s pomočjo katere ugotovimo kaj abstrakcija počne in kako to počne med implementacijo. Gre za zbirko operacij, ki opisujejo kako razred ali komponenta deluje. Sodelovanje je skupina razredov, vmesnikov in drugih elementov, ki delujejo skupaj v kooperativnem obnašanju, katerega rezultat je izboljšano delovanje sistema. Primer uporabe je opis niza zaporednih akcij, ki jih sistem izvaja z namenom, da določenemu akterju prinese rezultat, ki ga lahko opazujemo [100].
- Komponenta je fizični in zamenljivi del sistema, ki omogoča izvedbo niza vmesnikov. Vozlišče je fizični element, ki obstaja v času izvajanja programa in predstavlja programski vir, z lastno rezervacijo pomnilnika [100].
- Elementi obnašanja so dinamičen del UML modela in predstavljajo

časovno in krajevno obnašanje modela. Elemente obnašanja delimo na interakcije in diagrame stanj. Interakcija je obnašanje, ki vključuje nabor sporočil, ki se izmenjujejo med objekti v okviru določenega konteksta za doseganje določenega cilja. Diagram stanja je obnašanje, ki opisuje zaporedje stanj objekta ali interakcij med objekti, kot odgovor na dogodke v sistemu. Vključuje številne elemente stanja, prehode med posameznimi objekti, dogodki in aktivnosti [100].

- Element združevanje je organizacijski del UML, katerega glavni element imenujemo paket, ki omogoča združevanje elementov. Služi za združevanje modelnih elementov na nivoju arhitekturnega načrtovanja v večje skupine, s katerimi je lažje upravljati. Zadnji izmed elementov je element opomb, katerega uporabljamo za pojasnjevanje in opisovanje delovanja posameznih elementov v modelu [100].

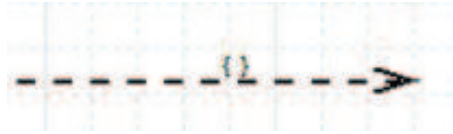
### Povezave

Kadar govorimo o povezavah, govorimo o spoju dveh elementov. V UML se pojavljajo štiri vrste povezav: odvisnosti, asociacije, generalizacije in realizacije.

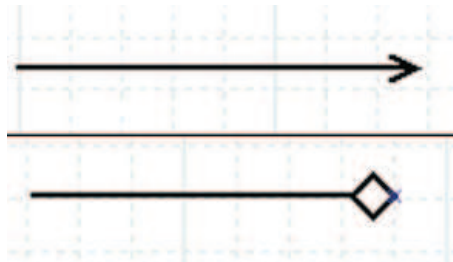
- Odvisnost je semantična povezava dveh elementov, pri katerih lahko sprememba neodvisnega elementa povzroči spremembo odvisnega elementa. Odvisnost se uporablja takrat, ko poizkušamo pokazati, da ena stvar uporablja drugo (glej Sliko 3.3).
- Asociacija je strukturirana povezava s katero pojasnujemo nabor vezi, katere so povezave med objekti. Kot enega izmed posebnih primerov lahko izpostavimo agregacijo, ki predstavlja povezavo med celoto in deli te celote (glej Sliko 3.4).
- Generalizacija je povezava, ki predstavlja povezavo specializacije in generalizacije, v kateri so objekti specializiranega elementa nadomestljivi z elementi generaliziranega elementa (glej Sliko 3.5).
- Realizacija je semantična povezava med elementoma, pri kateri eden element določi dogovor, ki ga mora drugi izvršiti. Le ta se pojavi med vmesniki v razredih ali med komponentami, ki jih realizirajo (glej 3.6).

### Diagrami

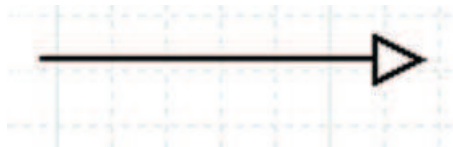
Diagram je vizualni model, ki predstavlja pogled na določenem nivoju abstrakcije. V UML ločimo strukturne diagrame, ki predstavljajo statično



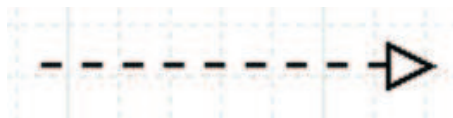
Slika 3.3: Odvisnost



Slika 3.4: Asociacija in agregacija



Slika 3.5: Generalizacija



Slika 3.6: Realizacija

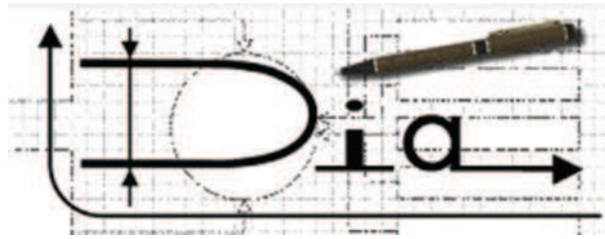
Strukturni diagrami	Diagrami, ki prikazujejo delovanje in uporabo	Diagrami, ki prikazujejo interakcije
Razredni diagram ( <i>Class Diagram</i> )	Diagram primerov uporabe ( <i>Use Case Diagram</i> )	Diagram zaporedja ( <i>Sequence Diagram</i> )
Objektni diagram ( <i>Object Diagram</i> )	Diagram aktivnosti ( <i>Activity Diagram</i> )	Komunikacijski diagram ( <i>Communication Diagram</i> )
Diagram komponent ( <i>Component Diagram</i> )	Diagram načrtovanja stanj ( <i>State Machine Diagram</i> )	Časovni diagram ( <i>Timing Diagram</i> )
Diagram razvrstitve ( <i>Deployment Diagram</i> )		Pregledni diagram interakcij ( <i>Interaction Overview Diagram</i> )
Strukturni diagram ( <i>Composite Structure Diagram</i> )		
Diagram paketov ( <i>Package Diagram</i> )		

Tabela 3.1: Vrste diagramov

strukturo sistema, diagrame, ki prikazujejo delovanje in uporabo ter diagrame, ki prikazujejo interakcije. Diagrami, ki prikazujejo delovanje in uporabo ter diagrami, ki prikazujejo interakcijo, predstavljajo tako imenovano dinamično strukturo sistema (Tabela 3.1).

## 3.2 O programskem orodju

Urejevalnik diagramov Dia (v nadaljevanju Dia) je odprtokodno programsko orodje za kreiranje tehničnih diagramov. Deluje na različnih operacijskih sistemih (Linux, Windows itd.). Je enostavno za učenje in dovolj prožno, saj uporabnikom omogoča ustvarjanje številnih diagramov. Programsko orodje uporabljamo v različnih industrijskih panogah. Primer: električar uporablja Dia za ustvarjanje diagrama, kjer prikaže delovanje vezja; programer ustvarja diagrame za prikaz izvedbe poti.



Slika 3.7: Urejevalnik diagramov Dia

### Prenos in namestitvev

Programsko orodje Dia lahko prenesemo in namestimo iz spletnega naslova Live.gnome [6] s klikom na razdelek *Download* na uvodni spletni strani. Pred prenosom lahko izbiramo med različnimi verzijami programskega orodja, odvisno kateri operacijski sistem uporabljamo. Uporabniki operacijskega sistema Ubuntu lahko programsko orodje Dia enostavno poiščejo, prenesejo in namestijo s pomočjo orodja Synaptic (glej Sliko 3.8).

Diagram je lahko sestavljen iz predmetov, ki so različnih oblik, črt, barv, velikosti itd. V razdelku *Orodja* se nahaja paleta vnaprej določenih predmetov, vključno s preprostimi oblikami, črtami in specifičnimi predmeti. Če želimo dodati določen predmet na platno, preprosto kliknemo na zeleni predmet in ga dodamo s klikom na miško. Po platnu premikamo predmete z miško in jih na to med sabo povezujemo (glej Sliko 3.9).

Dia vključuje nabor standardnih oblik in linij, ki so že vnaprej določene. Vključujejo diagram poteka, UML diagram, diagram omrežja, kronogram in mnoge druge diagrame. Glede na področje raziskovanja si lahko izberemo predmete, ki jih potrebujemo za oblikovanje diagrama.

Že izdelan diagram lahko shranimo v poljubno mapo v svojem računalniku. To storimo tako, da v orodni vrstici *Datoteka* kliknemo možnost *Shrani kot*. Odpre se novo okno, v katerem napišemo ime naše datoteke in jo shranimo na zeleno mesto (glej Sliko 3.10).

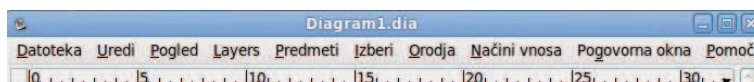
Da v nadaljevanju zagotovimo normalno shranjevanje in tiskanje dokumenta, v razdelku *Datoteka* izberemo možnost *Nastavitev strani*, kjer izbiramo med velikostjo papirja, usmerjenostjo, robovi in povečavo. Dia podpira izvoz v številne formate, kot so: *.cgm*, *.eps*, *.dia*, *.jpeg*, *.svg*, *.pdf*, *.jpeg ipd.*, katera izberemo iz orodne vrstice *Izvozi* (glej Sliko 3.11).

Ob zagonu Dia se ustvari prazno okno. V primeru, ko želimo novo okno za ustvarjanje v razdelku *Datoteka* izberemo možnost *Novo*. Odpre se novo polje za risanje diagramov. Osnovno programsko orodje Dia je sestavljeno iz dveh oken. V prvem oknu so zapisana programska orodja, za izdelavo diagramov

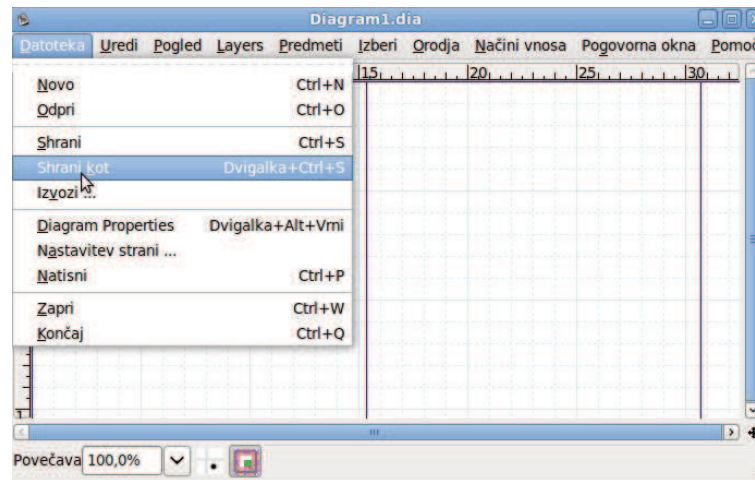




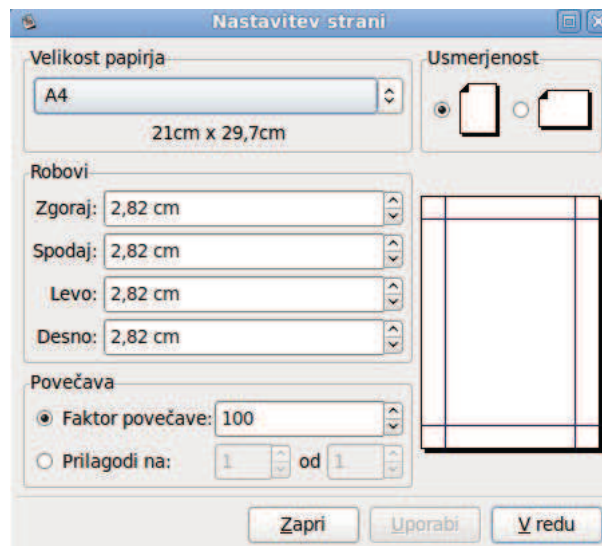
Slika 3.8: Prenos programskega orodja Dia



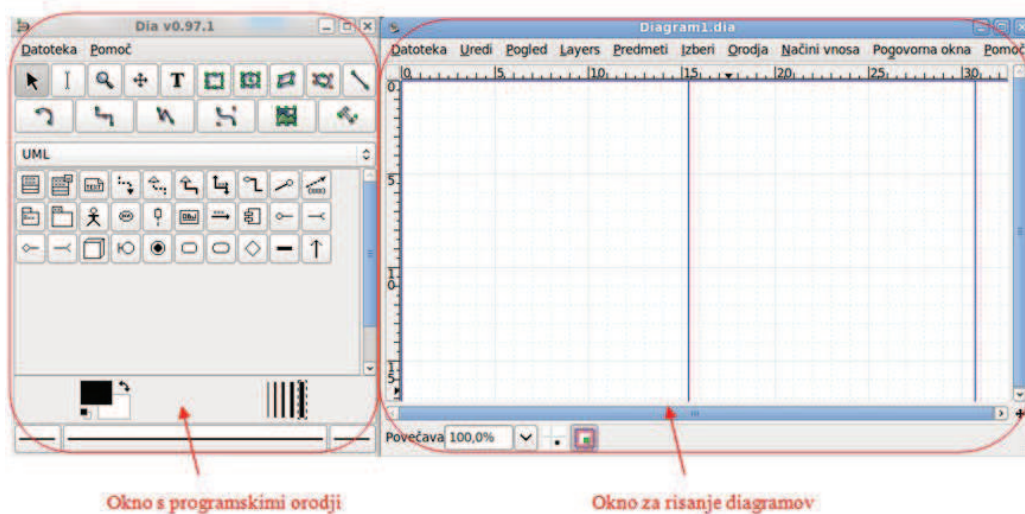
Slika 3.9: Menijska vrstica



Slika 3.10: Orodna vrstica - Datoteka



Slika 3.11: Nastavitev strani



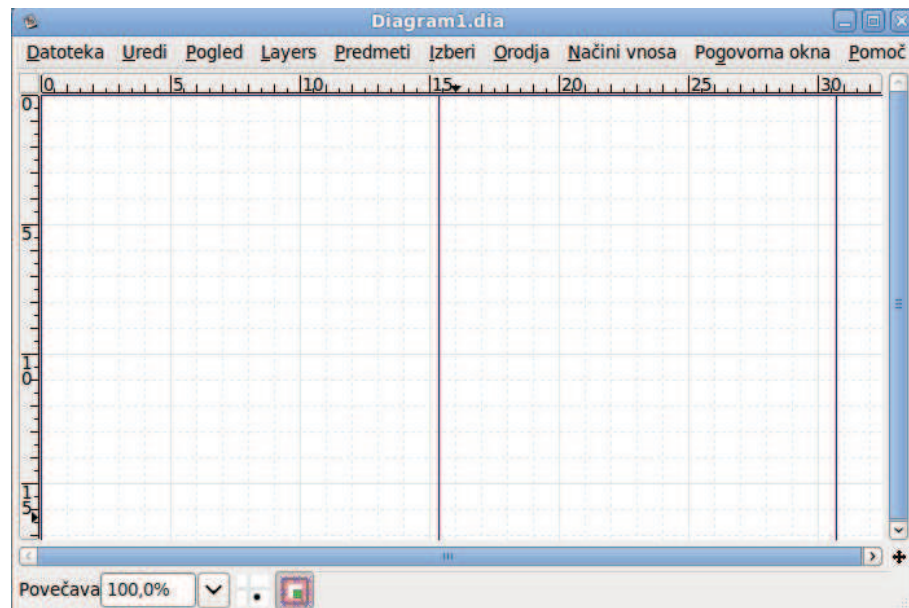
Slika 3.12: Prikaz osnovnih in specifičnih orodij

(okno s programskimi orodji). V drugem pa je prostor za oblikovanje le teh. Prvi sklop orodij uporabljamo za prilagoditev diagrama (spremeni predmet, urejanje besedila, povečava in drsenje po diagramu). Naslednje ikone so vgrajene za vstavljanje osnovnih predmetov (besedilo, škatla, elipsa, mnogokotnik, črta, slika). Srednji del predstavljajo specifična orodja, torej tista, ki smo jih izbrali iz seznama, glede na področje diagrama (npr. UML, diagram procesa...). Kvadrata, ki se nahajata pod specifičnimi orodji, omogočata nastavitve privzetih barv ospredja in ozadja za vse nove predmete diagrama ter debelin črt, ki jih lahko poljubno spreminjamo.

Slika 3.12 prikazuje osnovni okni, ki jih vidimo v Ubuntu. V podobni zgradbi se programsko orodje prikaže tudi v okolju Windows, le da okni nista ločeni - pojavljata se kot celota.

Na osnovnem oknu je prikazan osnovni meni, ki je namenjen oblikovanju diagrama. Uporabniku je omogočeno, da poveča pogled osnovnega okna maksimalno na 2500 % oz. da ga pomanjša minimalno do 5 %. Povečanje služi predvsem zato, da si lahko natančneje ogledamo dele diagrama. S pomanjšanjem vidimo velikost izdelanega diagrama in kaj je potrebno spremeniti, da ga lahko natisnemo oz. shranimo v pdf. obliko. Na osnovnem oknu je dodano še merilo (ravnilo), ki omogoča natančnejši izris diagrama (glej Slika 3.13).

V orodni vrstici *Pogled* (v okolju Windows) imamo možnost uporabe številnih funkcij, kot so: uporaba celotnega zaslona (F11), preklopi na mrežo in izključni



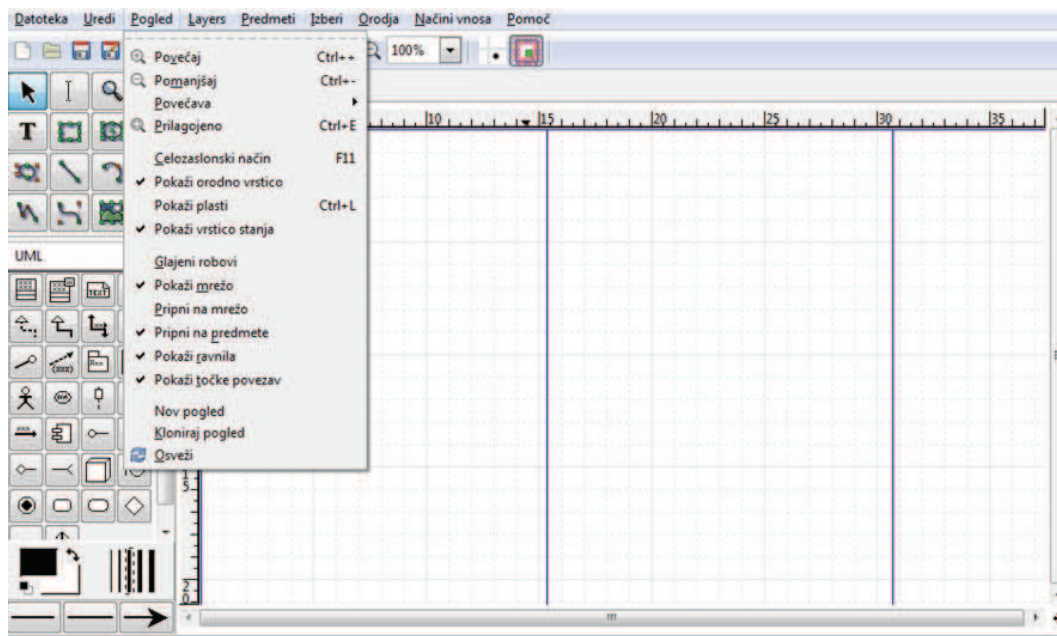
Slika 3.13: Osnovno okno

iz mreže, pokaži ravnilo, nov pogled, glajeni robovi itd. V okolju Ubuntu Linux je zapisanih manj možnosti, kar je razvidno tudi iz Slik 3.14 in 3.15.

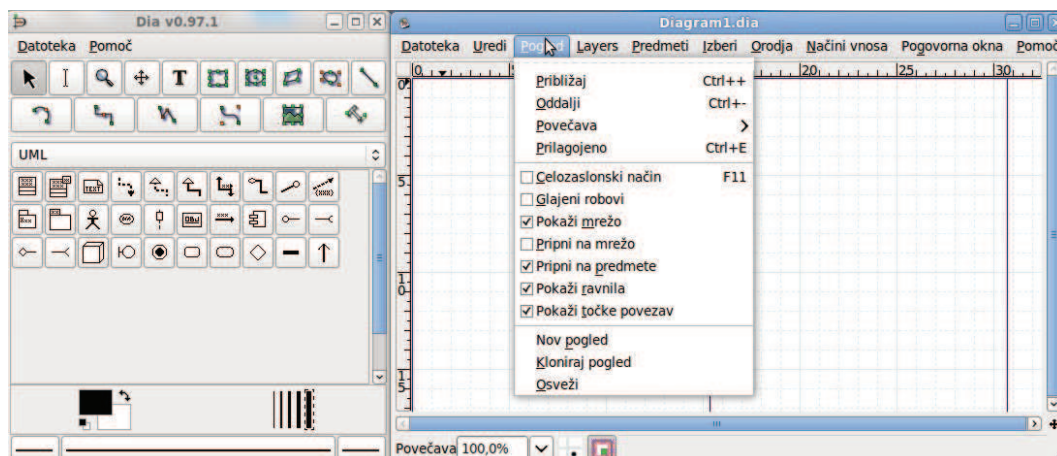
Če uporabnik oceni, da pri svojem delu potrebuje tudi druga orodja, kot so zapisana v osnovnih skupinah razvrščenih datotek, lahko to stori na preprost način. V razdelku *Datoteka* izberemo možnost *Listi in predmeti*, kjer označimo določen predmet, izberemo tipko *Kopiraj* in ga prenesemo v drugo kategorijo (glej Sliko 3.16).

Program DIA vsebuje 36 različnih oblik orodij iz najrazličnejših področij (glej Sliko 3.17).

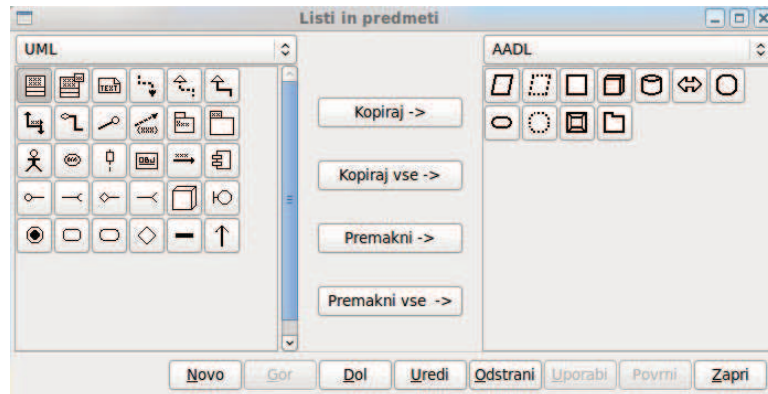
Na Sliki 3.18 je prikazan primer specifičnih standardnih UML orodij za prikaz diagrama aktivnosti ter ostala znana orodja (diagram poteka, strojništvo, luči...).



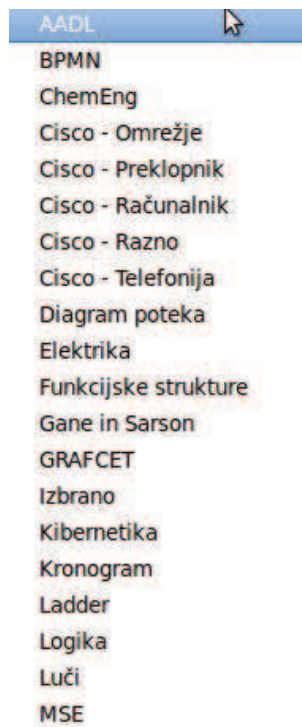
Slika 3.14: Orodna vrstica - Pogled (Windows)



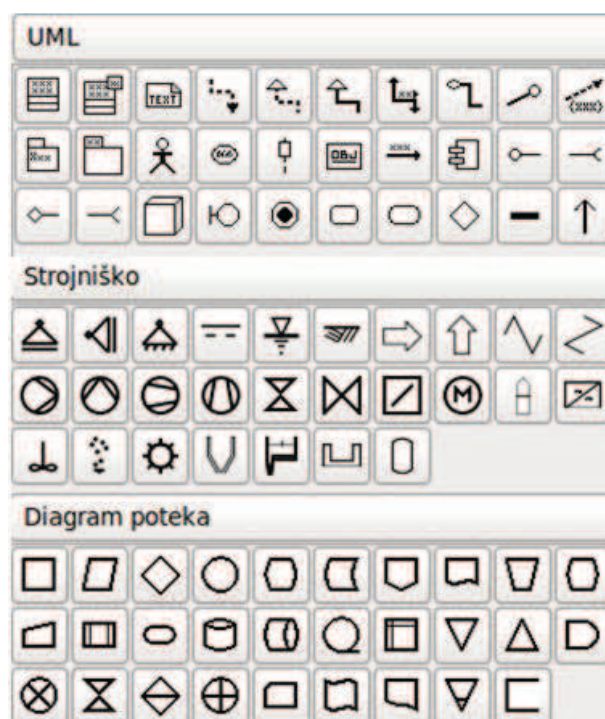
Slika 3.15: Orodna vrstica - Pogled (Ubuntu Linux)



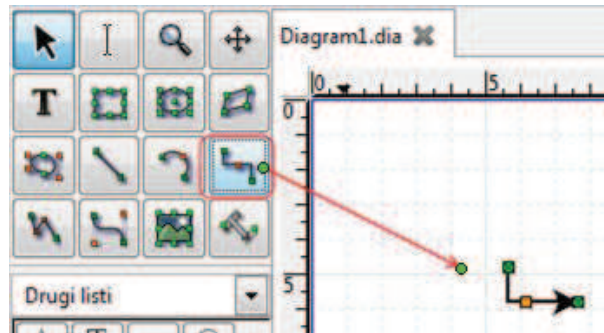
Slika 3.16: Listi in predmeti



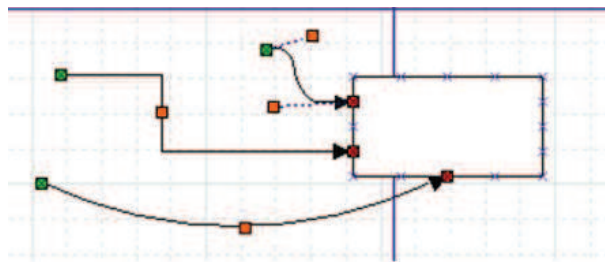
Slika 3.17: Možnosti oblik orodij



Slika 3.18: Specifična orodja



Slika 3.19: Dodajanje predmetov na platno



Slika 3.20: Oblikovanje in povezovanje predmetov

Dodajanje predmetov na platno se izvede s klikom na ikono predmeta, ki ga želimo dodati. Prenesemo ga iz orodij na osnovno okno, kjer ga nato oblikujemo. Če želimo predmet premakniti, kliknemo kjerkoli znotraj predmeta (ali nekje na črto) in ga z miško povlečemo na zeleno mesto (glej Sliko 3.19).

V številnih diagramih so oblike med seboj povezane z eno od osnovnih linij predmetov (oznaka "x"). Obstajajo pa tudi povezane točke v sredini vsake oblike, s pomočjo katerih lahko poljubno spreminjamo obliko predmeta oz. črte. Zelen kvadrček "x" omogoča, da linijo povežemo še z drugimi linijami (glej Sliko 3.20).

Besedilo vnašamo tako, da najprej izberemo predmet, ga prenesemo v polje risanja in kliknemo nanj ter vnesemo besedilo. Velikost pisave, poravnave in druge elemente, spreminjamo tako, da dvokliknemo na predmet (glej Sliko 3.21).

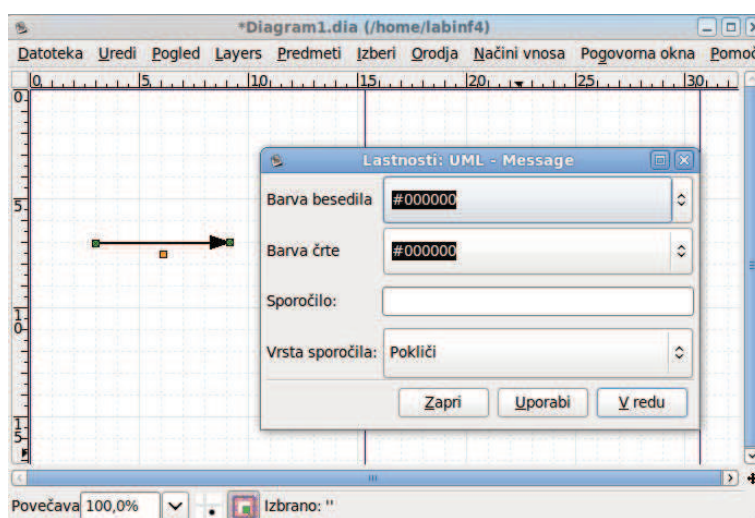
Program Dia vsebuje tudi možnost spreminjanja barv predmetov in črt. Barvo spremenimo tako, da z desnim klikom na miško, izberemo možnost lastnosti (glej Sliko 3.22).

Uporabnik lahko več o programskem orodju izve v razdelku *Pomoč*, ki se nahaja v menijski vrstici v orodnem oknu. Razdelek *Pomoč* omogoča hitro

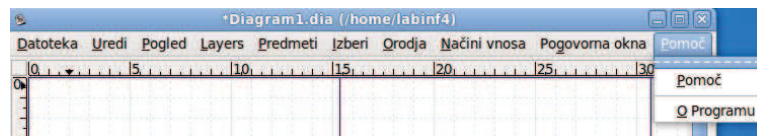




Slika 3.21: Dodajanje besedila v izbrani predmet



Slika 3.22: Lastnosti diagrama



Slika 3.23: Pomoč

seznanitev s potekom dela, v primeru, da naletimo na kakšno oviro ali pa če želimo zgolj izpopolniti svoje znanje. Pot do pomoči je prikazana na Sliki 3.23.

### Problem

V namišljenem podjetju se vsakodnevno srečujemo s procesi dela, ki jih je potrebno medsebojno usklajevati in dopolnjevati. Zavedati se je potrebno, da pravilno načrtovanje diagramov aktivnosti pripomore k učinkovitosti delovanja podjetja.

S programskim orodjem Dia prikažemo diagram aktivnosti v podjetju. Osredotočimo se na proces oskrbe, ki vključuje prikaz procesa naročanja, dostave, izmenjave dokumentov, grobega in finega prevzema platišč, reklamacij in skladiščenja. Smiselnost izdelave diagramov se pokaže predvsem v zahtevnejših primerih, ko aktivnosti ne moremo obravnavati ločeno, saj se medsebojno prepletajo in dopolnjujejo. V podjetju se vsakodnevno srečujemo s procesi dela, ki jih je potrebno medsebojno usklajevati in dopolnjevati. Zavedati se je potrebno, da pravilno načrtovanje diagramov aktivnosti pripomore k učinkovitosti delovanja.































## 3.3 Uporaba

Za prikaz danega problema preučimo diagram aktivnosti s standardnim jezikom modeliranja UML. Program Dia na Sliki 3.24 prikazuje predmete s katerimi izdelamo diagram aktivnosti. Na vsakem posameznem predmetu so zapisane njegove lastnosti in njegov pomen.

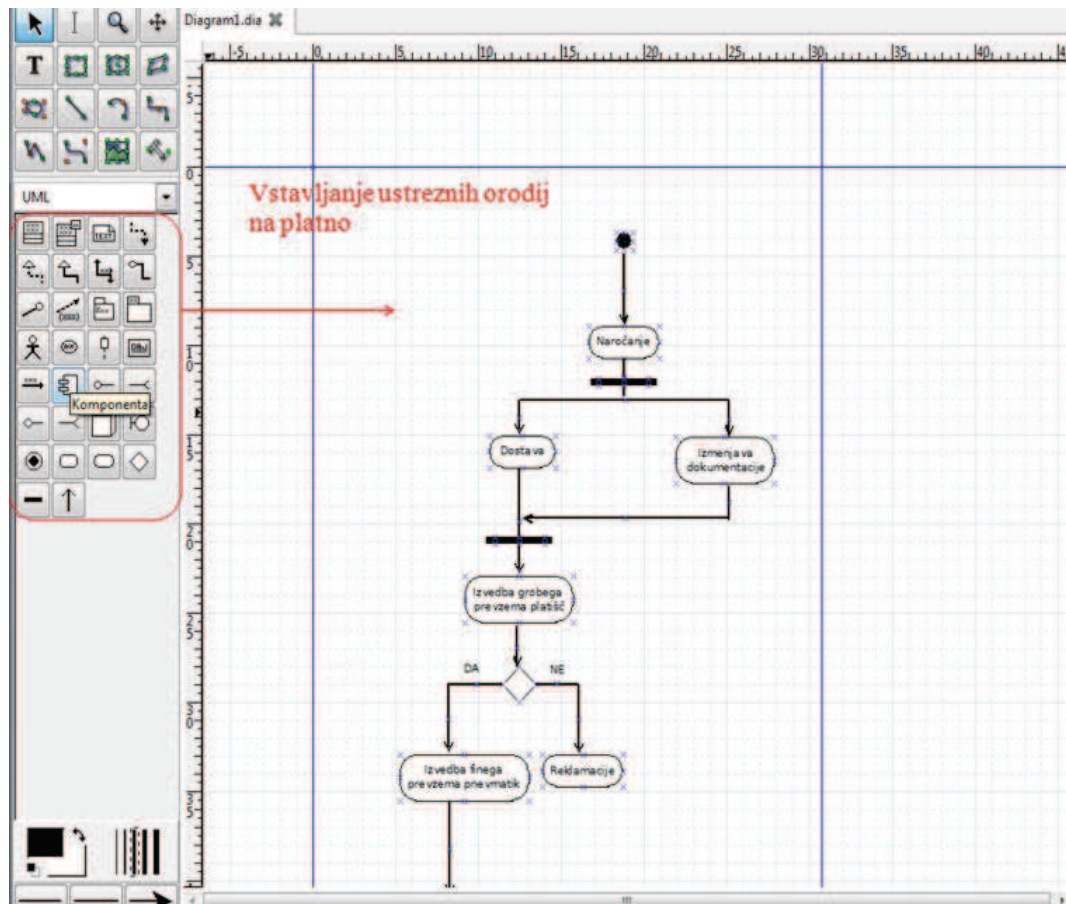
V nadaljevanju je prikazan preprost primer izdelanega diagrama z UML jezikom. Delo pričnemo tako, da na podlagi zastavljenega problema želimo narisati diagram aktivnosti za platišča, vse od naročanja, dostave, prevzema do skladiščenja.

Diagram aktivnosti je nazorneje prikazan na Sliki 3.26.

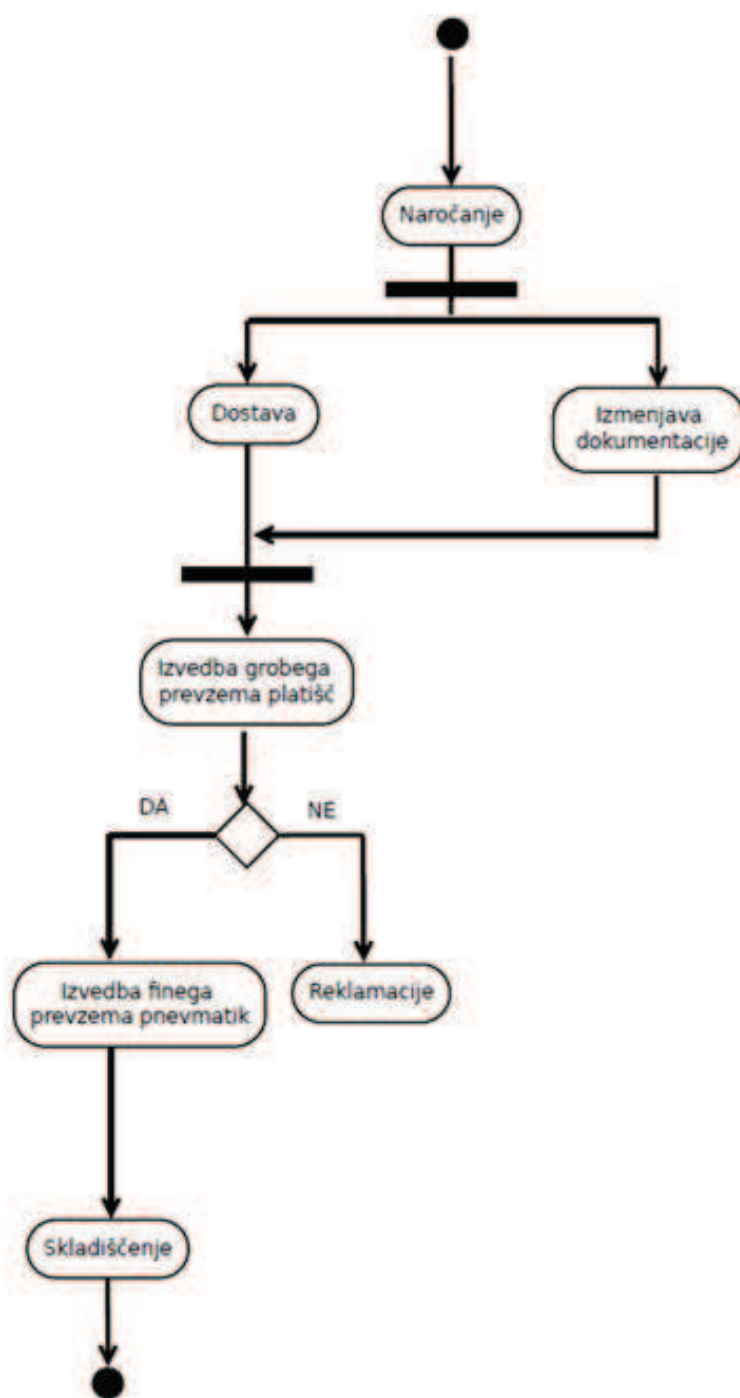
Dia ponuja možnost pogleda v tako imenovano *Drevo diagrama*, ki ga najdemo pod razdelkom *Datoteka*. Drevo diagrama je ena izmed možnosti,

 Razred	 Realizira, implementira določen vmesnik
 Posploševanje, delovanje	 Omejitev (dodaj omejitve na nekaj)
 Majhen paket	 Časovna linija UML
 Predmet	 Zbiralnik
 Razred predlog	 Dogodkovni odtok
 Povezava (dva razreda sta med sabo povezana)	 Ikona stereotipa razreda
 Velik paket	 Razvejaj
 Sporočilo	 Razvejaj/spoji
 Opomba	 Vozlišče
 Agregacija (en razred je del drugega)	 Začetno/končno stanje
 Igralec/izvršilec dejanja (dogodka)	 Vir dogodka
 Komponenta	 Pretvorba
 Odvisnost	 Stanje
 Implementiranje	 Aktivnost
 Primer uporabe	 Faseta

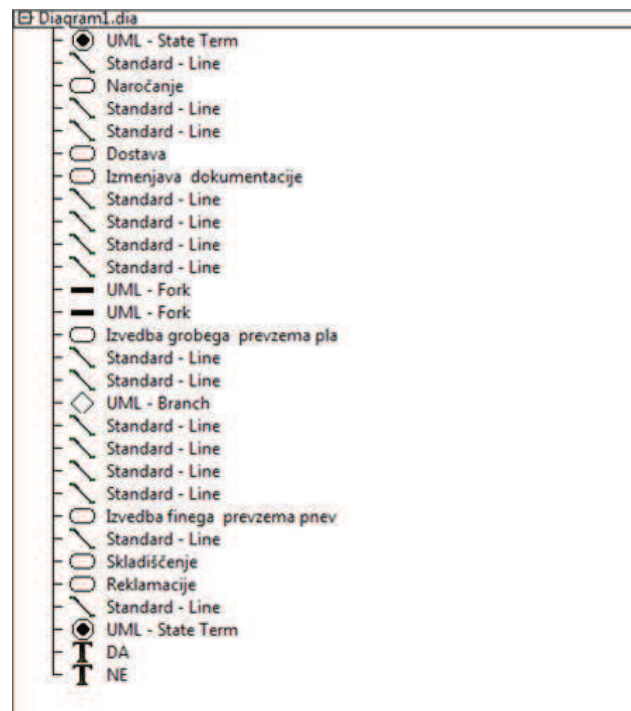
Slika 3.24: Orodja UML



Slika 3.25: Lastnosti diagrama



Slika 3.26: Diagram aktivnosti z UML



Slika 3.27: Drevo diagrama

ki prikazuje, kako je sam diagram sestavljen, kateri predmeti so vključeni in kaj le ti pomenijo (glej Sliko 3.27).

**Povzetek**

Diagram aktivnosti je ključnega pomena za planiranje in prikaz posameznih elementov ali celotnega dela projekta. S programskim orodjem Dia prikažemo diagram aktivnosti, ki se povsem razlikuje od diagrama, ki ga izdelamo s programskim orodjem Planner. Osredotočimo se na standard UML, na podlagi katerega deluje tudi Dia. Prikažemo proces oskrbe z izbrano komponento v podjetju (proces naročanja, dostave, izmenjave dokumentov, grobega in finega prevzema platišč, reklamacij in skladiščenja).

Z Dia imamo možnost izrisa najrazličnejših oblik tehničnih diagramov. Je uporabno programsko orodje za vsakega študenta, ki se pri študiju in izpopolnjevanju sreča s tovrstno problematiko. Prednost je v delovanju na različnih operacijskih sistemih, neplačljivost, lahka dostopnost, enostavnost učenja in prožnost.

