

## **3 KONCEPTI MODELOVANJA**

1

### **Modelovanje:**

- predstavlanje ili pojednostavljenje stvarnosti
- omogućava pravljenje nacrtu sistema
- uključuje elemente sa širokim efektima i izostavlja beznačajne na datom nivou apstrakcije

2

### **Zašto modelovati?**

- da bismo bolje razumeli sistem koji modelujemo
- da bismo obezbedili model strukture ili ponašanja sistema
- da bismo eksperimentisali istražujući višestruka rešenja
- da bismo uveli apstrakciju u upravljanje kompleksnošću
- da bismo dokumentovali projektne odluke
- da bismo vizuelizovali sistem 'kao što je' i 'biće'
- da bi se obezbedio šablon za konstruisanje sistema

3

### **Model #1:**

- Izbor modela ima ogromni uticaj na to kako se prilazi problemu i kako je rešenje uobličeno.
- Model može biti generisan na različitim nivoima apstrakcije (preciznosti).
- Efektni modeli moraju biti vezani sa stvarnošću .
- Nije dovoljan samo jedan model. Netrivijalni sistemi najbolje se opisuju skupom nezavisnih ali povezanih modela.

4

## Model #2:

Modeli se prave iz više razloga:

- Model je lakše i brže razviti.
- Model se može koristiti u simulacijama da bi se više naučilo o stvarima koje predstavlja.
- Model može da napreduje sa našim učenjem o stvari koju modeluje.
- Može se uvesti apstrakcija u model.
- Model može da predstavlja stvarne ili imaginarne stvari iz proizvoljnog domena.

5

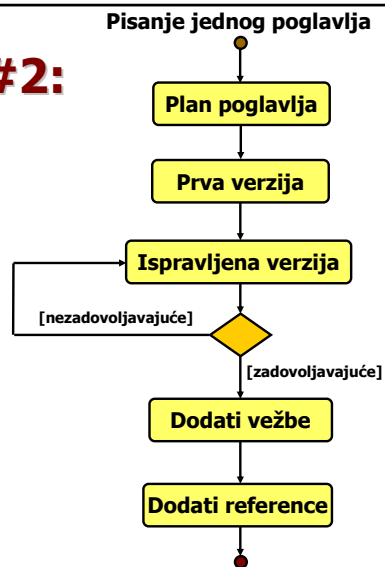
## Dijagram #1:

Projektanti koriste dijagrame za formiranje modela sa ciljem:

- razmene ideja,
- generisanja novih ideja i mogućnosti
- provere ideja i pravljenja predviđanja
- razumevanja struktura i odnosa.

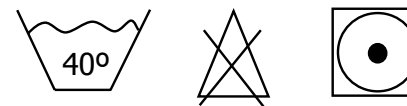
6

## Dijagram #2:



7

## Dijagram #3:



*Primer standarda dijagrama.*

8

## Razlika između modela i dijagrama:

- **Dijagram** može da ilustruje ili dokumentuje neki aspekt sistema.
- **Model** predstavlja kompletno gledište sistema na nekom posebnom nivou i sa posebne perspektive.

9

## Modeli i dijagrami:

- Model može da sadrži samo jedan dijagram i podatke i dokumentaciju kao podršku.
- Većina modela se sastoji od mnogo dijagrama jer je potrebno pojednostaviti kompleksne sisteme radi lakšeg razumevanja.

10

## Zašto učiti



- Dijagrami pomažu razumevanju poslovnog sistema ili softverskog posupka.
- Dijagrami olakšavaju prenošenje znanja.
- UML je postao preovlađujući grafički jezik.

11



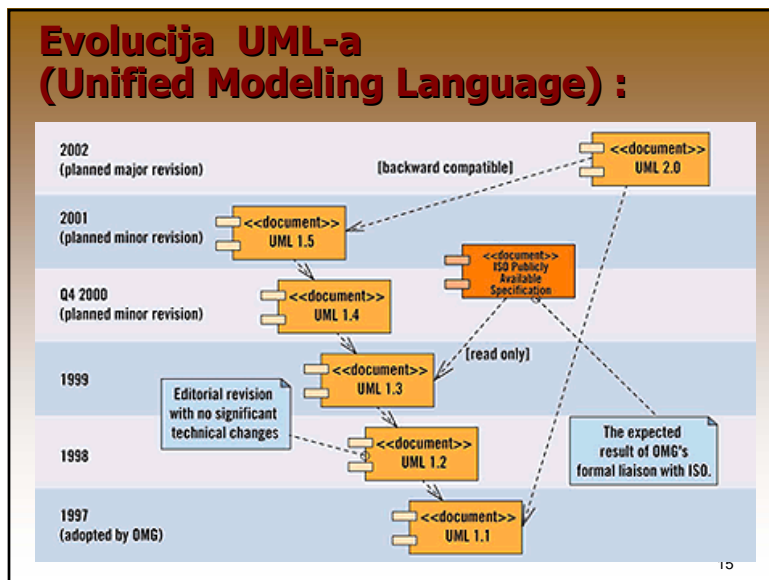
*UML je postao  
preovladujući grafički  
jezik.*

13

### Kratka istorija UML-a:

- 1) 1996.godine je Jacobson ujedinio metod Booch-a i Rumbaugh-a da bi dobio UML 0.9
- 2) UML Partneri radili su sa Amigos-om da bi OMG-u predložili UML kao standardni jezik za modelovanje 1996.
- 3) U 1997, UML partneri ponudili su svoj inicijalni predlog (UML 1.0) OMG-u i 9 meseci kasnije podneli su finalni predlog (UML 1.1).

14



### Šta je UML?

- UML je jezik za vizualizaciju, specifikaciju, konstrukciju i dokumentovanje artefakta intenzivno softverskih sistema.
- Primeri artefakta: zahtevi, arhitektura, projekat, izvorni kod, testni slučajevi, prototipovi, itd.
- UML je pogodan za modelovanje raznih vrsta sistema: privredni informacijski sistemi, distribuirane web-aplikacije, ugrađeni real-time sistemi, itd.

16

## UML – jezik za specifikaciju:

- Obezbeđuje gledišta za razvoj i razvijanje UML ne zavisi od procesa
- Preporučuje se za upotrebu sa procesima koji su:
  - a) pobuđivani korisničkim slučajevima
  - b) arhitekturno- centriranih
  - c) iterativni
  - d) inkrementacioni

17

## Ciljevi UML-a:

- obezbediti modelarima spreman za upotrebu, izražajni i vizuelni jezik za modelovanje da bi razvijali i razmenjivali značajne modele,
- obezbediti mehanizme proširivosti i specijalizacije da bi se proširili suštinski koncepti,
- podržati specifikacije koje su nezavisne od posebnih programskih jezika i razvojnih procesa,
- obezbediti formalnu osnovu za razumevanje jezika specifikacije,
- podsticaj/rast tržišta objektnih alata,
- podržati više nivoe razvoja sa konceptima kao što su okviri za komponente ili moduli.

18

## Modeli u UML-u:

- **Sistem** je predmet koji se modeluje.
- **Podsistem** je deo sistema koji se sastoji od povezanih elemenata.
- **Model** je apstrakcija sistema ili podsistema sa određenog gledišta.
- **Dijagram** je grafička predstava skupa elemenata u modelu sistema.

19

## Gledišta UML-a:

Booch et al. (1999):

Pet gledišta korišćenih uz UML:

- gledište korisničkog slučaja
- gledište projektovanja
- gledište procesa
- gledište implementacije
- gledište razvijanja

20

## **Paketi (packages):**

**Paketi** su način organizovanja i grupisanja elemenata.

Upotreba kod CASE alata

(kao način upravljanja modelima koji se proizvode).

21

## **Razvijanje modela:**

Modeli koji se prave u toku razvoja sistema menjaju se sa napredovanjem projekta. Menjaju se po tri dimenzije:

- apstrakcija,
- formalnost i
- nivo detalja.

22

## **Dijagrami aktivnosti:**

- modelovanje zadatka (napr. u poslovnom modelovanju);
- opisivanje funkcije sistema predstavljenog pomoću upotrebnog slučaja;
- specificiranje operacija, za opis logike operacije;
- u **USDP** (Unified Software Development Process) **ZA** modelovanje aktivnosti koje čine životni ciklus.

23

## **Notacija kod dijagrama aktivnosti:**

- Skup aktivnosti povezanih prelascima sa jedne na drugu aktivnost.
- Svaka aktivnost predstavlja se pravougaonikom sa zaobljenim uglovima unutar kojeg se piše ime aktivnosti.

24

## Dijagram aktivnosti, primer#1:

*(Dve aktivnosti udružene prelaskom)*



25

## Dijagram aktivnosti, primer#2:

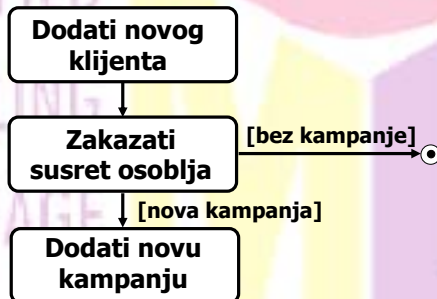
*(Aktivnosti sa tačkom odluke)*



26

## Dijagram aktivnosti, primer#3

*(Izbor predstavljen bez tačke odluke)*



27

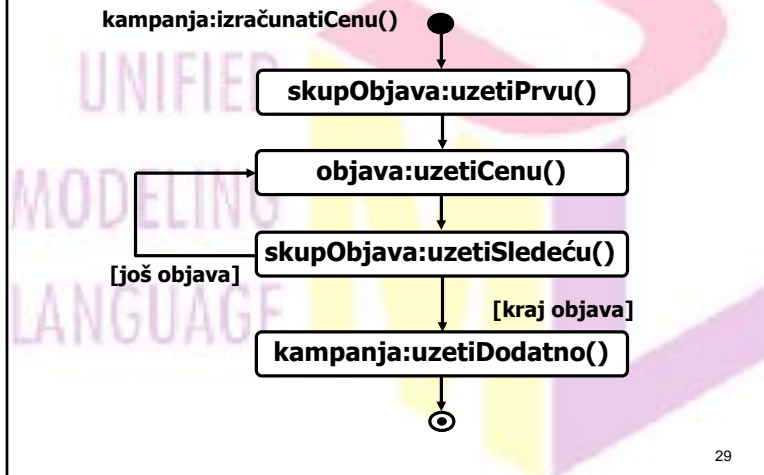
## Objekti u dijagramima aktivnosti:

Dva načina za prikaz objekata u dijagramima aktivnosti:

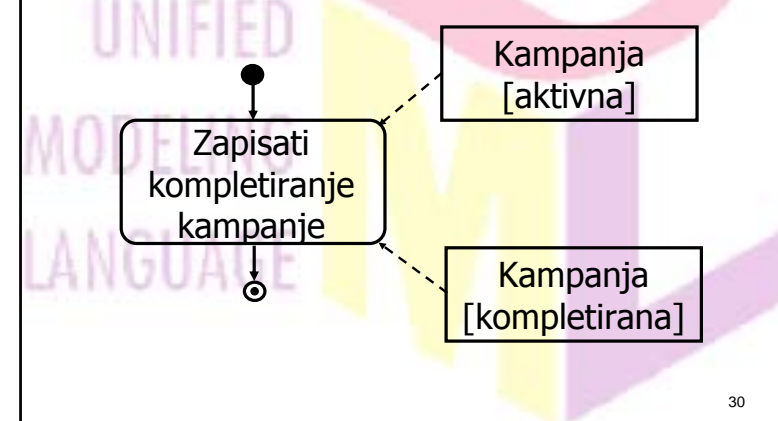
- 1) oznaka operacije nekog objekta može se koristiti kao ime aktivnosti;
- 2) objekat se može prikazati kao obezbeđivanje ulaza ili izlaza neke aktivnosti.

28

## 1) Dijagram aktivnosti sa oznakama operacija objekata kao aktivnostima:

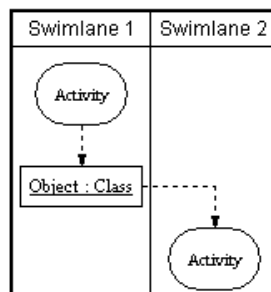


## 2) Dijagram aktivnosti sa tokom objekata:



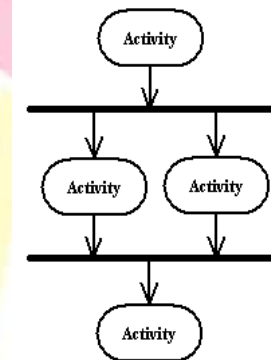
## Dijagram aktivnosti sa plivačkim stazama:

Srodne aktivnosti smeštaju se u istu kolonu nazvanu *plivačka staza (swimlane)*.



## Sinhronizacija:

- Sinhronizaciona linija pomaže ilustrovanju paralelnih prelaza.
- Sinhronizacija: račvanje i spajanje





## Glavne aktivnosti:

- Prikupljanje i modelovanje zahteva
- Analiza zahteva
- Projektovanje sistema
- Projektovanje klasa
- Projektovanje interfejsa
- Projektovanje upravljanja podacima
- Konstrukcija
- Testiranje
- Implementacija

33



34

