

Uvod v modele

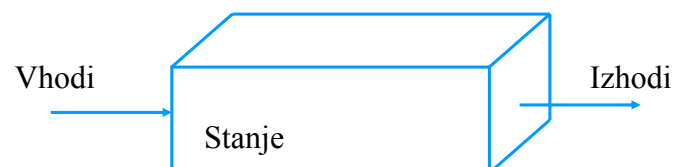
N. Zimic

N. Zimic

10-1

Sistem

- V teoriji sistemov ločimo:
 - strukturo sistema (notranjo zgradbo sistema),
 - obnašanje sistema.
- Model sistema:

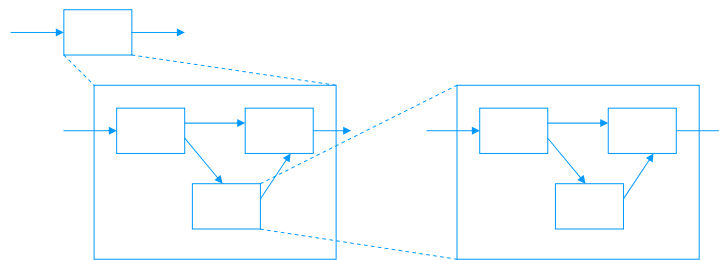


N. Zimic

10-2

Dekompozicija sistema

- Dekompozicija se uporablja za razdelitev sistema na manjše dele, ki so lažje obvladljivi.
- Pri postavitvi modelov si pomagamo s kompozicijo, ki je obraten proces in kjer iz manjših elementov zgradimo velik sistem.



N. Zimic

10-3

Primerjava z objektnim programiranjem

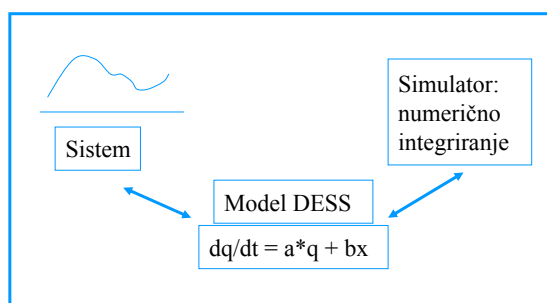
- Sisteme se lahko primerja z objekti, ki se uporabljajo pri objektnem programiranju.
- Poglavitne razlike med sistemi in objekti so:
 - objekti običajno niso hierarhični in modularni kot sistemi,
 - sistemi so formalne strukture, ki delujejo na osnovi časa,
 - objekti niso enostavno povezljivi v smislu vhoda in izhoda.

N. Zimic

10-4

Model DESS

- Model DESS (Differential Equation System Specification) je model, kjer so relacije med vhodi in izhodi opisane z diferencialnimi enačbami:

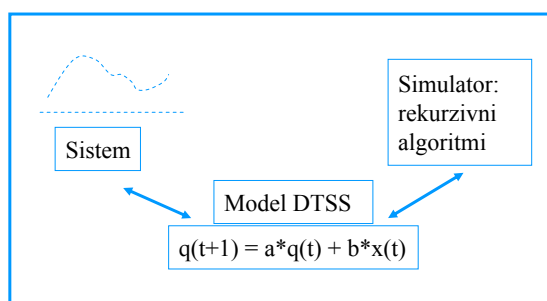


N. Zimic

10-5

Model DTSS

- Model DTSS (Discrete Time System Specification) je model, kjer so relacije med vhodi in izhodi opisane z enačbami, ki so opredeljene v diskretnem času:

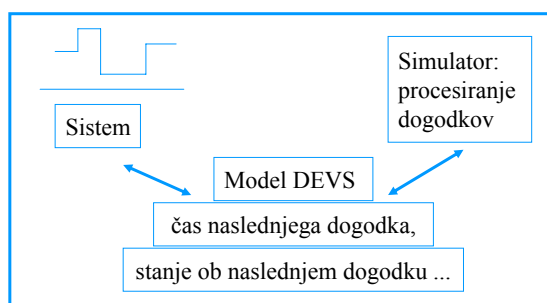


N. Zimic

10-6

Model DEVS

- Model DEVS (Discrete Event System Specification) je model, kjer dogodki pogojujejo spremembo stanja sistema:

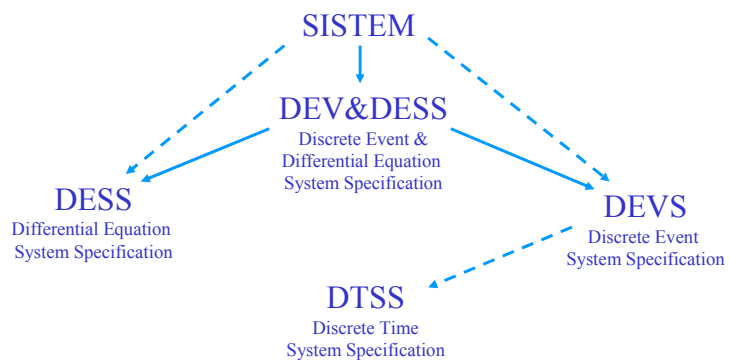


N. Zimic

10-7

Združitev formalnih modelov

- Model DEVS in DESS se lahko združi v enoten formalni model DEV&DESS:

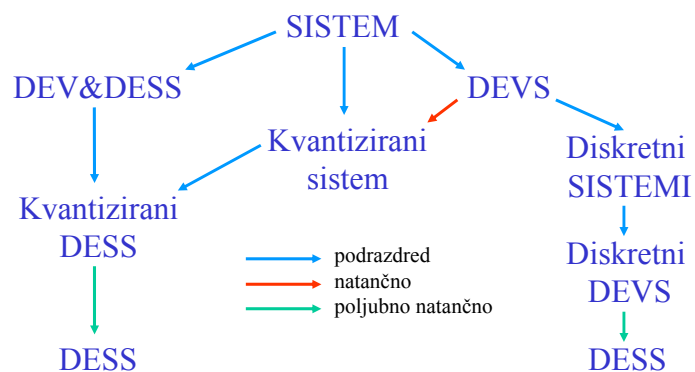


N. Zimic

10-8

Kvantizirani formalni modeli

- Kvantizacija modelov je potrebna predvsem zaradi računalniške simulacije:



N. Zimic

10-9

Hierarhična ureditev

- Specifikacija sistema je hierarhično urejena:

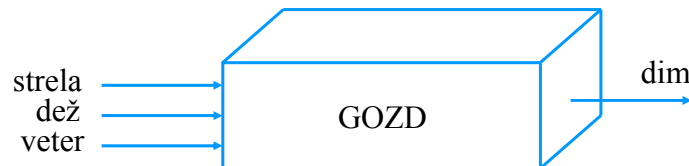
0. opazovanje sistema,
1. vhodno/izhodne značilnosti,
2. vhodno/izhodne funkcije,
3. prehajanje stanj,
4. povezovanje komponent.

N. Zimic

10-10

Opazovanje sistema

- V opazovanju sistema je določeno, katere spremenljivke zajemamo in kako jih opazujemo skozi čas.
- Kot primer si oglejmo gozdni požar, kjer kot vhodi nastopajo: strela, dež in veter, kot izhod pa dim. Seveda bi lahko kot izhod nastopale tudi druge spremenljivke, kot na primer temperatura.

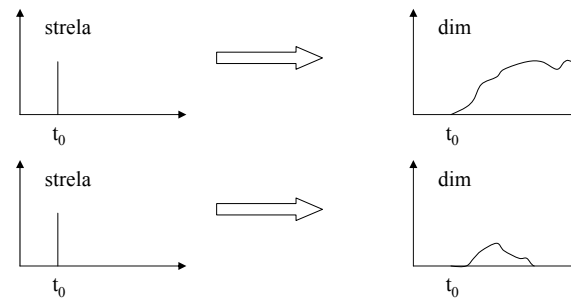


N. Zimic

10-11

Vhodno/izhodne značilnosti

- Na tem nivoju zajemamo podatke na vhidih in izhodih. Na osnovi zajemanja pridemo do podatkov, ki so funkcije časa. Pri istih vhodnih podatkih se lahko sistem različno odzove, kot je primer na spodnji sliki:



N. Zimic

10-12

Vhodno/izhodne funkcije

- Na tem nivoju se uvedejo začetni pogoji. Na osnovi začetnih pogojev se lahko ločijo odgovori sistema, ki so bili prikazani na prejšnji sliki. Zgornji primer je bil pri suhem gozdu, spodnji pa pri vlažnem gozdu.

N. Zimic

10-13

Prehajanje stanj

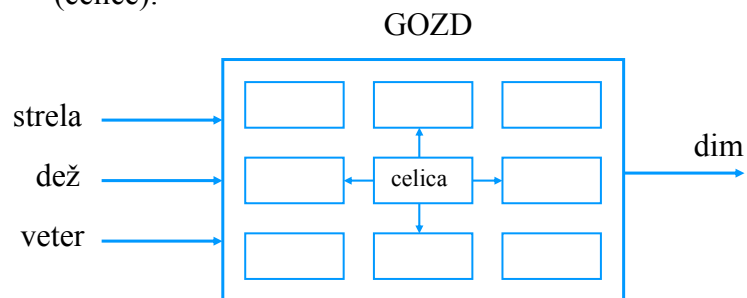
- Na tem nivoju je določeno, kako stanja med seboj prehajajo.
- Primer:
 - druga strela v suhem gozdu ne povzroči novega požara, ker je gozd že pogorel,
 - druga strela v mokrem gozdu povzroči požar, ker je gozd po prvi strela delno zgorel in se pri tem posušil.

N. Zimic

10-14

Povezovanje komponent

- Na tem nivoju se lahko komponente med seboj povezujejo in tvorijo večji sistem.
- Primer gozda, ki ga razdelimo na posamezna področja (celice):



N. Zimic

10-15

Vrste analiz

- Sistemska analiza
 - Sistem, ki ga analiziramo lahko obstaja ali pa ga načrtujemo. V obeh primerih želimo spoznati njegove značilnosti.
 - Ker imamo dostop do vseh podatkov, lahko analizo izvajamo od zgoraj navzdol. To pomeni, da začnemo z opazovanjem komponent in nadaljujemo z nižjimi nivoji.

N. Zimic

10-16

Vrste analiz (nad.)

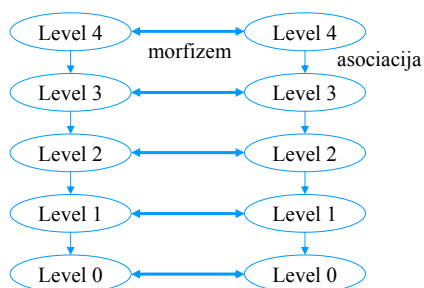
- Sistemska interferenca
 - Sistem, ki ga analiziramo obstaja in z vplivanjem na vhodne parametre želimo ugotoviti njegovo obnašanje.
 - Analizo izvajamo od spodaj navzgor, kar pomeni, da začnemo z opazovanjem sistema, naslednji korak pa so vhodno/izhodne značilnosti.

Vrste analiz (nad.)

- Sistemsko načrtovanje
 - Sistem, ki ga načrtujemo, še ne obstaja.
 - Analizo izvajamo od spodaj navzgor, kar pomeni, da začnemo z opazovanjem sistema, naslednji korak pa so vhodno/izhodne značilnosti.
 - Podatke generiramo in jih simuliramo.

Morfizmi

- Pri načrtovanju modelov vedno iščemo relacije med sistemskimi opisi:



N. Zimic

10-19

Morfizmi (nad.)

- Relacija med sistemom in sistemskim opisom:

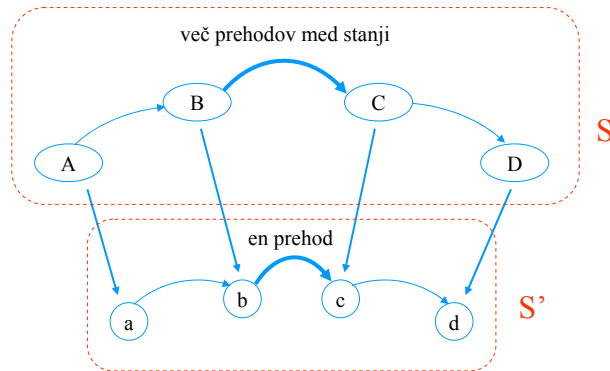
Nivo	Specifikacija sistema	Morfizem velja če:
0	opazovanje sistema	V relaciji so vhodi, izhodi in časovne odvisnosti
1	vhodno/izhodne značilnosti	Morfizem velja na nivoju 0 in vhodi in izhodi so v relaciji ena ena
2	vhodno/izhodne funkcije	Morfizem velja na nivoju 0 in stanja funkcij so v relaciji
3	prehajanje stanj	Razloženo na naslednji prosojnici
4	povezovanje komponent	Posamezne komponente so v relaciji

N. Zimic

10-20

Morfizmi (nad.)

- Primer prehajanja stanj (homomorfizem):

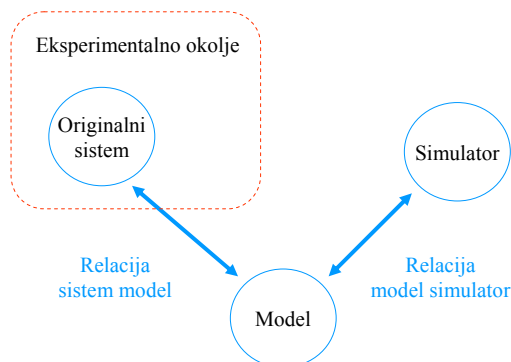


N. Zimic

10-21

Sestav modeliranja in simulacij

- Relacije:



N. Zimic

10-22

Eksperimentalno okolje

- Eksperimentalno okolje je spisek pogojev, pod katerimi se sistem opazuje, ali pa se z njim eksperimentira.
- V eksperimentalnem okolju se odraža namen, ki vodi k modelu in tudi k simulaciji.
- Isti model ali simulacija se lahko uporabljata v različnih okoljih.
- Prav tako se lahko različna okolja uporabljajo na istem modelu ali simulatorju.

N. Zimic

10-23

Eksperimentalno okolje (nad.)

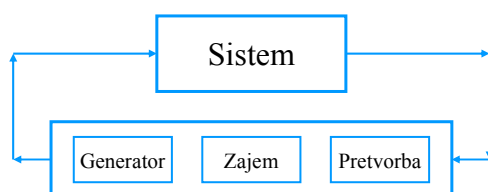
- Obstajata dva enakovredna pogleda na eksperimentalno okolje:
 - kot definicija podatkovnih tipov, ki se shranjujejo v podatkovno bazo,
 - kot sistem, ki je povezan z opazovanim sistemom in služi za pridobivanje potrebnih informacij pod določenimi pogoji

N. Zimic

10-24

Eksperimentalno okolje (nad.)

- Eksperimentalno okolje in njeni sestavni deli
 - generator služi za generiranje zahtev,
 - v zajemu se shranjujejo podatki,
 - modul za pretvorbo je namenjen pretvorbi vhodnih signalov v primerno obliko.

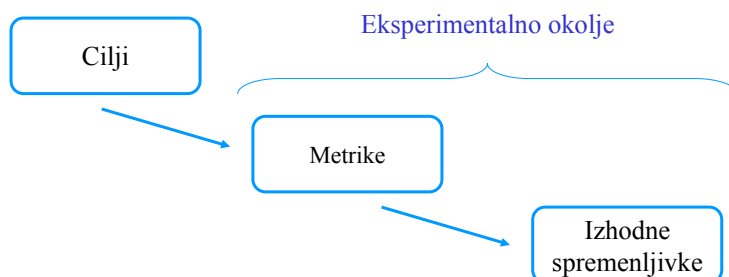


N. Zimic

10-25

Eksperimentalno okolje (nad.)

- Osnova je vedno natančno določen namen (cilj).
- Ko je določen namen, lahko postavimo metrike.
- V zadnjem koraku določimo izhodne spremenljivke.



N. Zimic

10-26

Model

- V splošnem je model formalni zapis na katerem koli nivoju.
- Običajno se pri MIS, kot model, upošteva formalni zapis najvišjih dveh nivojev.
- Model lahko vsebuje: instrukcije, pravila, enačbe ali funkcijo, ki opisuje vhodno izhodne značilnosti.
- Na tretjem nivoju opišemo model s prehajanjem stanj, ki sprejema vhodne podatke v času in generira izhodna zaporedja.
- Na četrtem nivoju opišemo model z elementi, ki izhajajo iz tretjega nivoja.

N. Zimic

10-27

Simulator

- Množica instrukcij in pravil potrebuje nek agent, ki bo sposoben izvajati instrukcije in pravila ter generirati odvisnosti.
- Agent je lahko računalnik, lahko pa je tudi človek.
- Simulator je običajno grajen na visokem nivoju in z posameznimi gradniki sestavljamo simulacijski program.

N. Zimic

10-28

Ločevanje modela in simulatorja

- Ločevanje modela in simulatorja ima kar nekaj prednosti:
 - isti formalni model se lahko izvaja na različnih simulatorjih, s čimer dosežemo prenosljivost modela,
 - za formalne modele lahko postavimo simulacijske algoritme, ki vsebujejo tudi rigorozno formalno preverjanje,
 - potrebne simulacijske zmogljivosti nam lahko služijo kot mera za kompleksnost simulacije.

Relacije

- Entitete (sistem, eksperimentalno okolje, model, simulator) postanejo pomembne, ko se med njimi opredeli ustrezna relacija.
- Če na primer zgradimo model sistema za točno določen namen, je ta model uporaben samo za ta namen in noben drug.

Preverjanje pravilnosti

- Relacija pravilnosti se uporablja med eksperimentalnim okoljem, sistemom in simulacijo.
- Pravilnost se preverja samo v območju, za katerega je model ali simulator napravljen.

N. Zimic

10-31

Potrditev simulatorja

- Simulator je potrjen, če se navzven obnaša kot model.
- Potrditev je potrebno izdelati na drugem nivoju sistema.
- Ker je simulator kar slika modela, je možno potrditi simulator tudi s preverjanjem relacij med gradniki (homomorfizem).

N. Zimic

10-32

Poenostavitev

- Poenostavitev modela je pogosto potrebna zaradi omejenih resursov, ki so na razpolago za izvajanje simulacije.
- Navkljub poenostavitvam, pa mora simulator še vedno dajati pravilne rezultate.