

Automatizacija procesa

Stepeni automatizacije

UVOD

Osnovno pitanje upravljanja (automatizacije) procesa jest kako definisati ciljeve upravljanja:

- zašto upravljati? (što očekujemo od upravljanja)
- čime upravljati? (kojim procesnim veličinama)
- kako upravljati? (odabrati odgovarajuću koncepciju upravljanja)

Upravljanje (automatizacija) procesa treba osigurati:

- sigurnost rada procesa,
- ekonomičnost,
- održavanje procesa unutar zadatih tehnoloških ograničenja,
- održavanje količine i kvaliteta proizvoda

UVOD

AKVIZICIONO-UPRAVLJAČKI SISTEMI

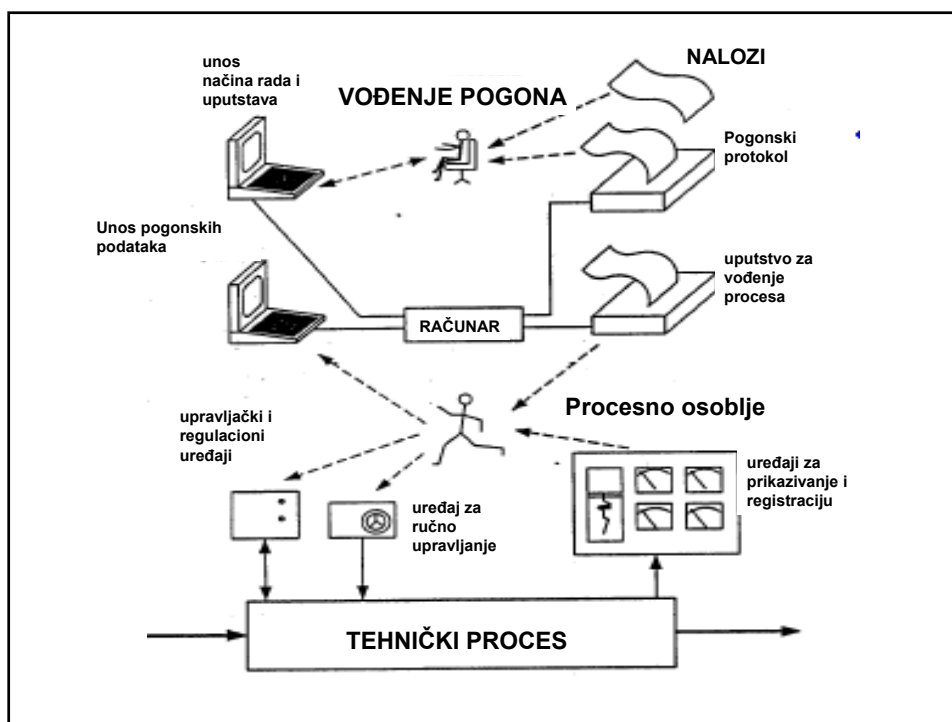
- Pod pojmom automatizacije i upravljanja procesima se često podrazumijevaju, pored lokalne automatike
 - kojom se obezbjeđuje automatski rad određenih uređaja,
- **akviziciono-upravljački sistemi** za automatsko prikupljanje relevantnih podataka vezanih za funkcionisanje tehnoloških procesa, obradu prikupljenih podataka i upravljanje na osnovu njih. Često se automatizovanim sistemima nazivaju i čisto akvizicioni sistemi koji posjeduju samo sposobnost automatskog prikupljanja, odnosno prikupljanje i obrade podataka

UVOD

- ❖ Pri projektovanju sistema za automatizaciju procesa treba načiniti analizu procesa u cilju određivanja dijelova procesa koje je smisleno automatizovati,
- ❖ Obzirom na stepen automatizacije definisani su sljedeći nivoi automatizacije:
 - Off-line rad s veoma malim stepenom automatizacije,
 - On-line rad u otvorenoj petlji sa srednjim stepenom automatizacije,
 - On-line rad u zatvorenoj petlji s visokim stepenom automatizacije.

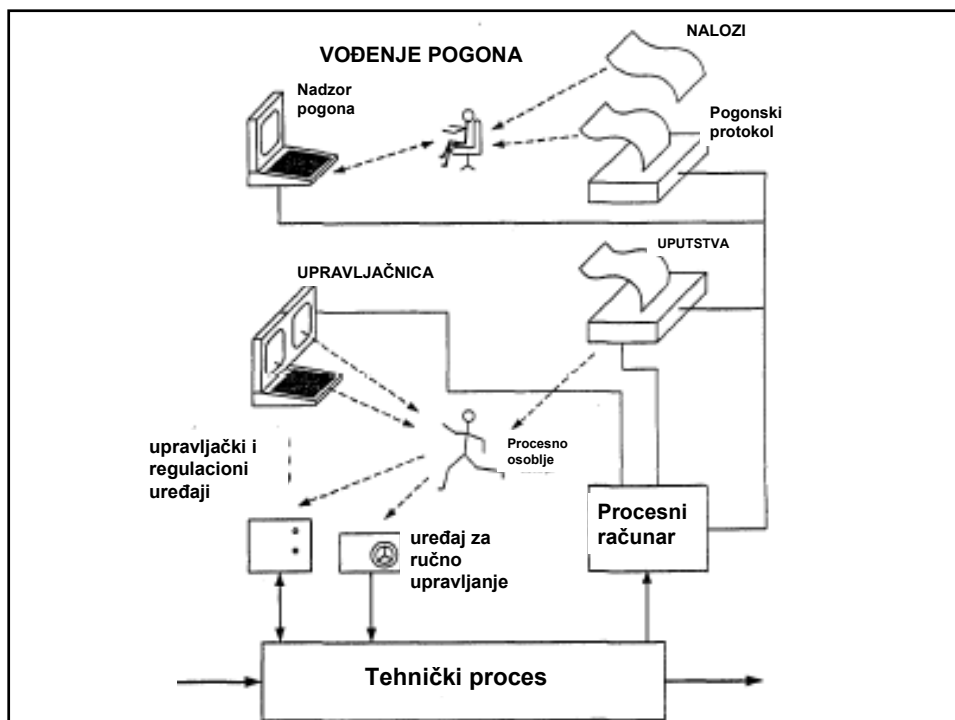
Off-line rad

- ❖ Vodenje procesa prepušteno je isključivo procesnom osoblju, tako da off-line način rada ne povećava stepen automatizacije procesa.
- ❖ Računar obavlja određene proračune i obrade. Nije direktno uključen u proces, pa se u ovom slučaju ne radi o procesnom računaru.
- ❖ Unos i izlaz podataka obavljaju se na načine uobičajene za računare opšte namjene.



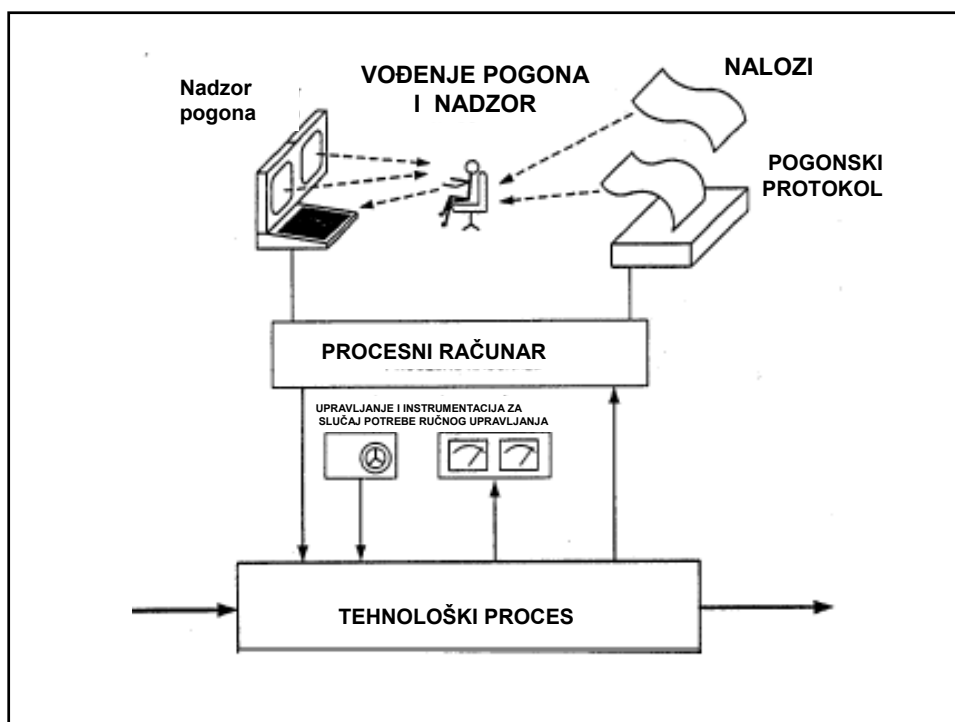
On-line rad u otvorenoj petlji

- ❖ Vodenje procesa takođe je prepušteno procesnom osoblju, a računar se koristi za praćenje stanja procesa. Informacije o stanju procesa prenose se u računar u stvarnom vremenu. Obavlja se samo operacija akvizicije podataka iz procesa.
- ❖ Za razliku od off-line rada osoblje dobija kvalitetnije informacije i uputstva za upravljanje procesom.
- ❖ Pri radu u otvorenoj petlji upravljanje sistemom takođe se temelji na iskustvu osoblja. Tendencija je da se iskustveno znanje ugradi u upravljački sistem za rad u zatvorenoj petlji => na znanju zasnovani sistemi.

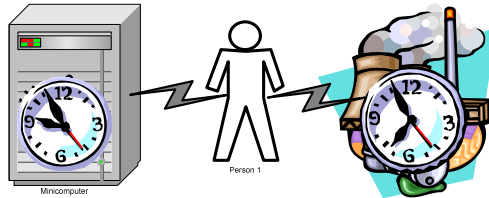


On-line rad u zatvorenoj petlji

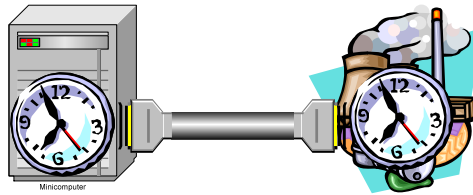
- ❖ Procesni računar na temelju trenutnog i ciljnog stanja procesa proračunava upravljačke veličine i neposredno djeluje na proces.
- ❖ Ovakav način rada postavlja visoke zahtjeve na procesni računar.
- ❖ Procesni računar treba da obavi i funkcije vezane za procesnu sigurnost, o čemu treba voditi posebnu brigu u tzv. sigurnosno-kritičnim tehničkim procesima (nuklearne elektrane, letjelice, željeznicki saobraćaj).



Računarski sistem za rad u realnom vremenu

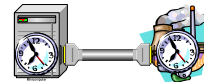


računar kao sredstvo za obradu podataka



računar koji radi u realnom vremenu

Dva tipa sistema za rad u realnom vremenu

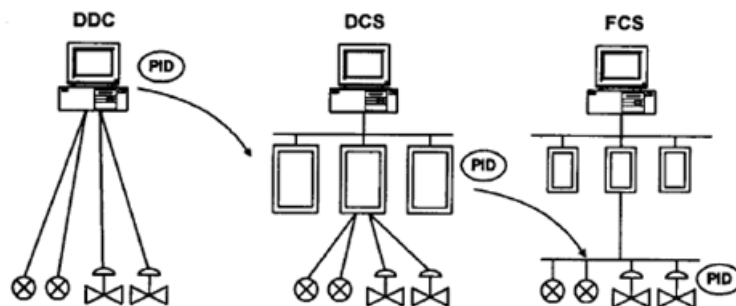


sistemi kod kojih se svaka grupa operacija mora završiti u okviru specificiranog maksimalnog vremena



sistemi kod kojih je srednje vrijeme izvršavanja operacija, mjereno na nekom definisanom obimu posla, manje od zadatog maksimalnog vremena

Evolucija upotrebe računara u AUS (akviziciono upravljački sistemi)



- ~ direktno računarsko upravljanje (DDC – Direct Digital Control)
- distribuirani upravljački sistemi (DCS - Distributed Control System)
- upravljački sistemi u polju (FCS - Field Control System)

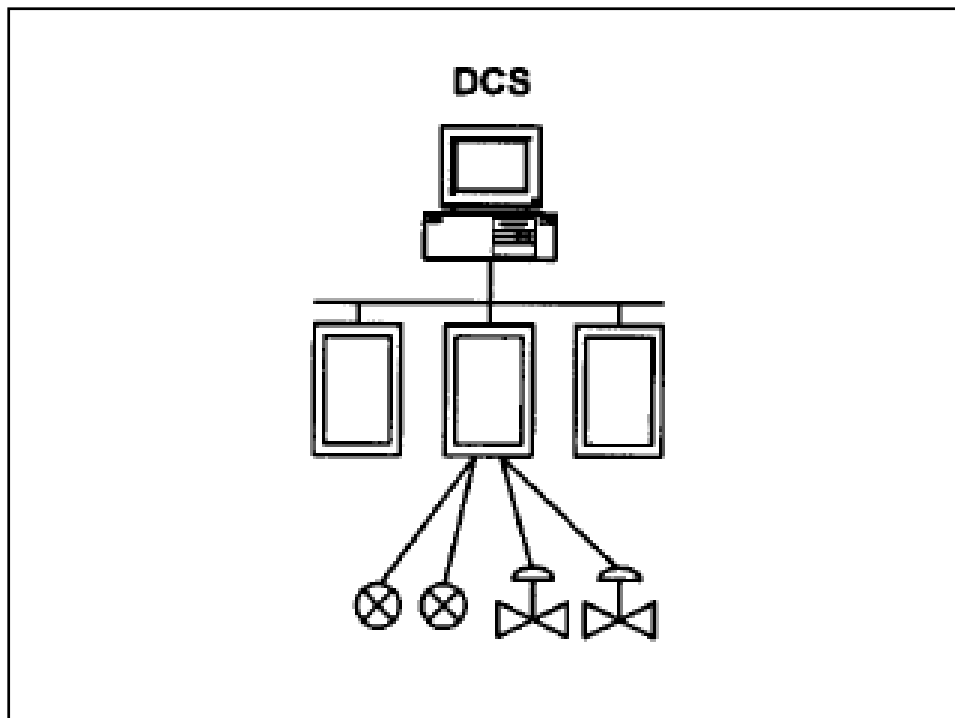
Direktno računarsko upravljanje (DDC – Direct Digital Control)

- sve prednosti digitalne obrade informacija (mogućnost izvođenja komplikovanih algoritama upravljanja, arhiviranje i lak pregled podataka, fleksibilnost upravljačkih algoritama, itd.).
- javio se problem da upravljanje sa nekoliko stotina takvih petlji, uz korišćenje samo jednog računara, dovodi do niske pouzdanosti rada sistema.
- u ovakvim sistemima jedan par žica se koristio za povezivanje uređaja u polju sa I/O (ulazno/izlaznom) karticom upravljačkog računara što je značajno podizalo cenu instalacije.



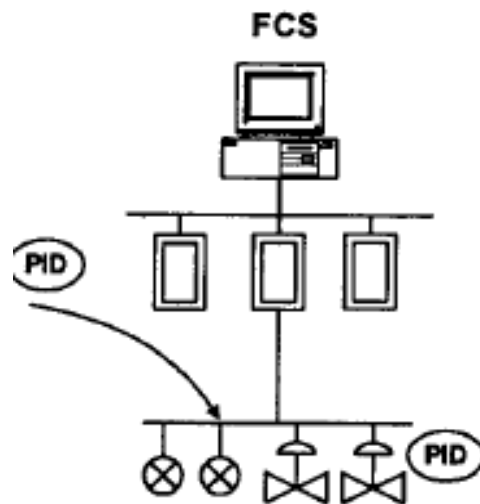
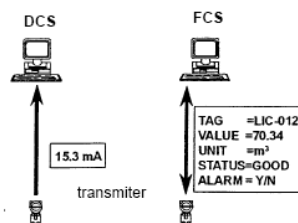
Distribuirani upravljački sistemi (DCS - Distributed Control System)

- Uvođenje digitalnih akviziciono-upravljačkih jedinica koje su postavljene na nivou procesa (procesne jedinice) i koje mogu međusobno, kao i s hijerarhijski višim nivoima upravljanja, da komuniciraju digitalno.
- Procesne jedinice čine PLC (engl. Programmable Logic Controller-PLC) uređaji i udaljene telemetrijske jedinice (engl. Remote Telemetry Unit - RTU) na kojima se izvršava najveći deo aplikativne upravljačke podrške.
- Svaka procesna jedinica je zadužena za akviziciju podataka sa više senzora i upravljanje sa više upravljačkih petlji, pri čemu je veza sa instrumentacijom (transmiteri, pozicioneri, ...) najčešće ostvarena žično, analognim strujnim signalom.
- Hijerarhijski viši nivo upravljanja u principu vrši vizualizaciju procesa kojim se upravlja i nužnu komunikaciju sa ljudskom posadom (pokretanje i zaustavljanje rada, promena režima rada, podešavanje raznih parametara i sl)

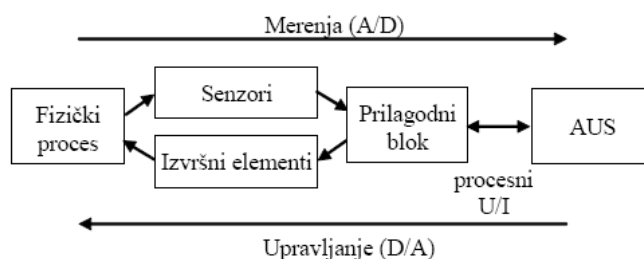


Upravljački sistemi u polju (FCS - Field Control System)

- Druga generacija distribuiranih upravljačkih sistema
- Podrazumeva primenu računara (CPU-jedinice) u samim instrumentima (senzorima, transponderima, aktuatorima, i sl.)
- Upravljačka logika spuštena na najniži mogući nivo, a komunikacija digitalnim protokolom ostvarena u čitavom sistemu.
- Za razliku od prve generacije DCS, gde je otkaz jedne kartice na PLC-u mogao da dovede do ispada iz funkcije nekoliko upravljačkih petlji, kod FCS otkaz uređaja dovodi do otkaza samo jedne upravljačke petlje.
- Korišćenje "pametnih" (engl. smart) aktuatora i transpondera



Struktura povezivanja akviziciono-upravljačkog sistema sa fizičkim procesom



Procesni ulazi

- Nivo kontinualnog električnog signala (strujnog ili naponskog) na **analognim ulazima** AUSA u svakom trenutku je proporcionalan trenutnoj vrijednosti mjerene fizičke veličine. Tipični opsezi ulaznih električnih signala su: 4-20 mA, $\pm 5V$, 0-10 V, 0-100 mV i sl. Na ovaj način se mjere pritisak, temperatura, masa i sl. Digitalizacija analognih ulaznih signala vrši se korišćenjem A/D konvertora. U praksi je očigledna tendencija da se digitalizacija električnog signala vrši već u sklopu samog pretvarača, korišćenjem tzv. "smart" transmitera.

- Za **brojačke ulaze** je karakteristično da učestanost električnih impulsa koji se prihvataju predstavlja mjeru trenutne vrijednosti fizičke veličine. Impulsni pretvarači se najčešće koriste za mjerenja protoka tečnosti ili gasova, mjerenje brzine rotacije (tahogenerator), ugla zakretanja i sl. Frekvencija generisanih impulsa u praksi je najčešće manja od 1 KHz, a njihov naponski nivo ne veći od 24 V jednosmjerne struje. Pošto se u okviru AUS-a impulsi signali prihvataju posredstvom digitalnih brojačkih kola, ovi procesni ulazi se nazivaju brojačkim ulazima.

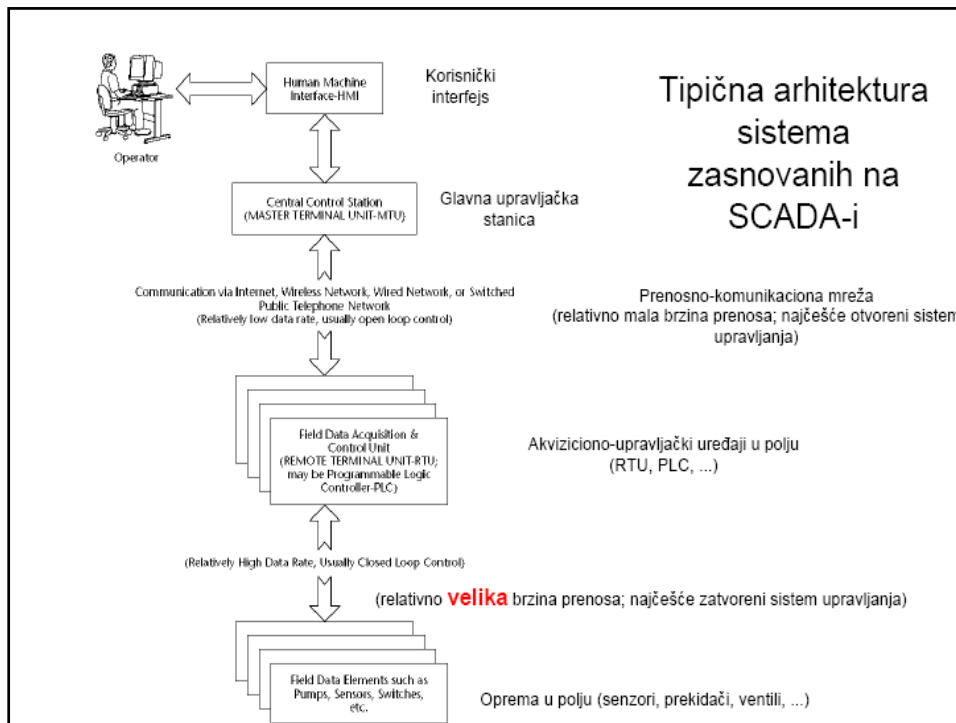
- **Digitalni ulazi** predstavljaju fizičke veličine koje su, već po svojoj prirodi, diskretne. Najčešće se preko njih prati stanje izvršnih elemenata u postrojenju, kao što su ventili (otvoren/zatvoren) ili sklopke (uključena/isključena). Izvor ovakvih signala su i razni graničnici (puno/prazno, max/min, itd.), sigurnosni prekidači poput presostata (nadpritisak/normalno), i sl. Ulazni električni signal je naponski (24Vdc ili 220Vac).

Procesni izlazi

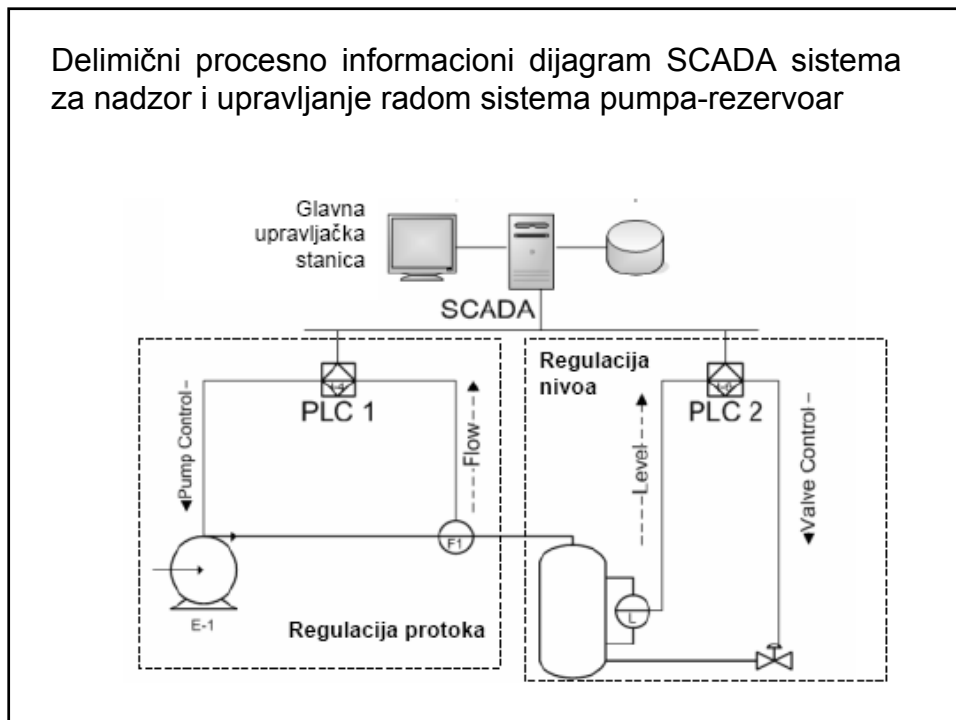
- Posredstvom digitalnih izlaza ostvaruje se upravljanje tipa uključi/isključi ili otvori/zatvori, za uređaje kao što su sklopke, kontaktori, aktuatori ventila i sl.
- Digitalni izlazi se najčešće opisuju radnim naponom i maksimalnom strujom koju mogu da propuste (24Vdc/1A, 220Vac/3A, i sl.).
- Analognim (kontinualnim) pobudnim signalom zadaje se radna tačka izvršnog elementa (pozicija regulacionog ventila, referentna vrijednost eksternog PID regulatora i sl).
- Analogni procesni izlaz opisuje se električnim opsegom izlaznog signala (0-20mA, 0-5 V, itd.).
- Generisanje analognog izlaznog signala vrši se korišćenjem D/A konvertora. Poput analognih transmitera, sve je veći broj tzv. "smart" izvršnih elemenata koji se, zahvaljujući ugrađenom mikroprocesorskom bloku, serijski povezuju sa akviziciono-upravljačkim sistemom. Ovakvi uređaji često preuzimaju izvršenje složenijeg lokalnog upravljanja (npr. PID regulacija), čime značajno pojednostavljaju izvođenje AUS-a, uz istovremeno podizanje ukupne raspoloživosti upravljačkog sistema.

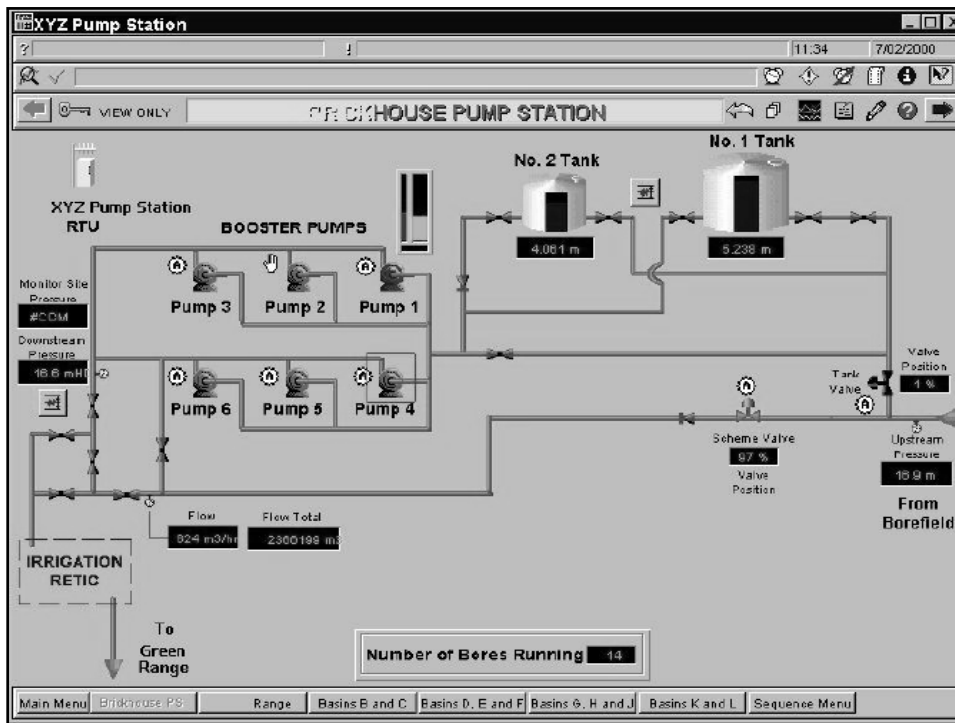
AKVIZICIONO-UPRAVLJAČKI SISTEMI

- Sistemi zasnovani na SCADA sistemima (engl. Supervisory control and data acquisition system)
- Industrijski akviziciono upravljački sistemi.
- obje klase akviziciono – upravljačkih sistema su izuzetno srodne i granica između njih nije sasvim očigledna, pogotovo je nejasna u delu njihove systemske programske podrške.

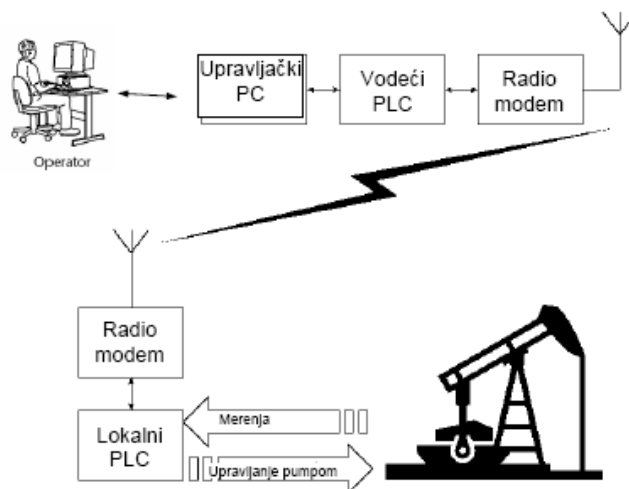


Delimični procesno informacijski dijagram SCADA sistema za nadzor i upravljanje radom sistema pumpa-rezervoar

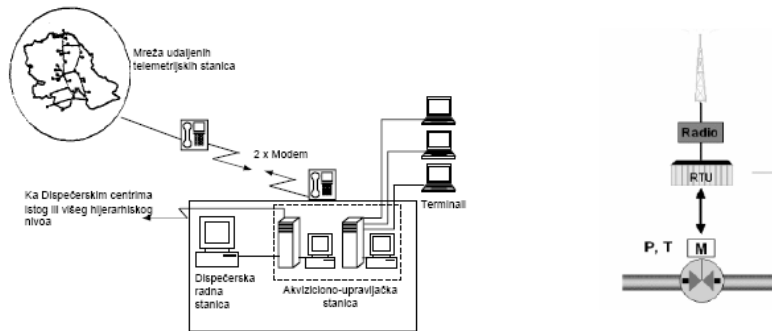




AUS za praćenje i upravljanje radom bušotine sa dubinskom pumpom



Telemetrijski SCADA sistema za akviziciju i upravljanja transportnim sistemom gasa

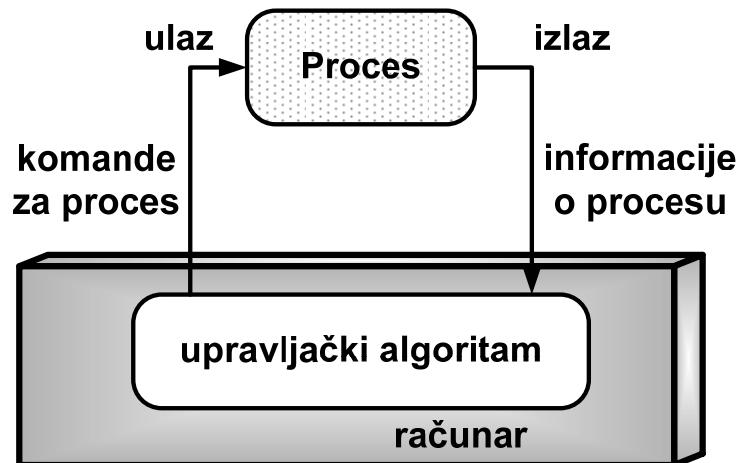
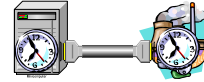


udaljene telemetrijske stanicama (engl. Remote Telemetry Unit - RTU) -mikroprocesorska stanica čija je primarna uloga sakupljanje mjerenih signala sa senzora i generisanje pobude izvršnih elemenata, odnosno konverzija procesnih veličina u digitalni oblik i obrnuto.

Industrijski akvizicijono upravljački sistemi

- upravljanje prostorno ograničenim industrijskim postrojenjima, odnosno kontrola tehnoloških operacija u okviru proizvodnih procesa
- manja dislokacija pojedinih elemenata sistema, pouzdanije odvijanje komunikacije, i visok stepen automatizacije upravljačkih aktivnosti
- Upravljački algoritam koji se ugrađuje u industrijske AUS sisteme je često vrlo složen, vremenski strogo uslovljen, i obuhvata automatizovano (programsko) vođenje čitavog proizvodnog procesa

Računarski upravljani sistem



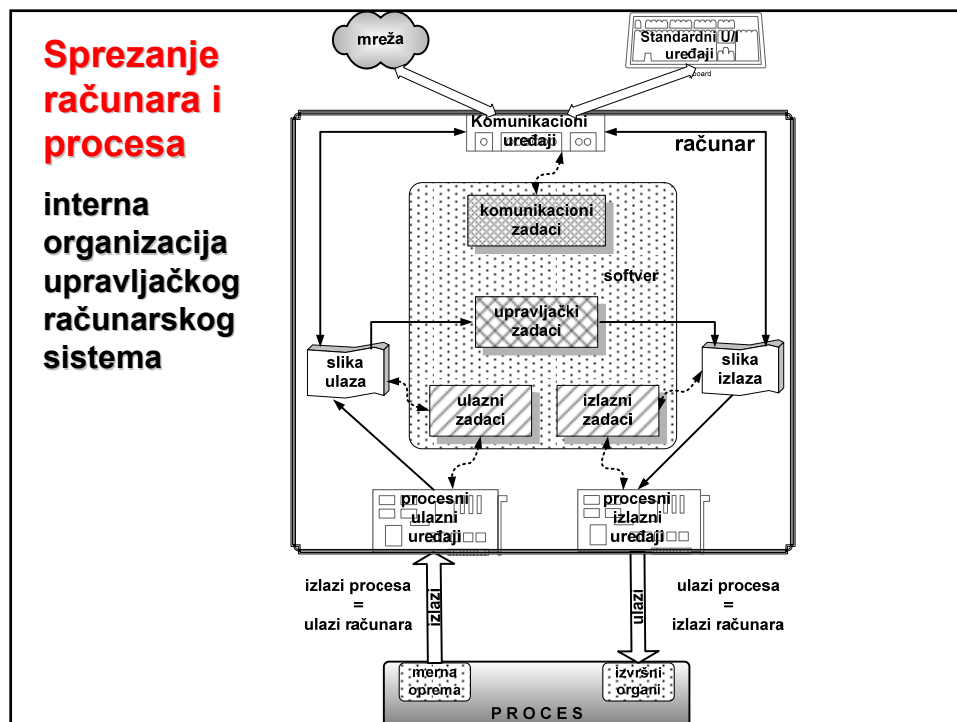
SPREZANJE PROCESA I RAČUNARA

- Svaki sistem automatskog upravljanja ima dvije cjeline – *proces* i *upravljanje*. U principu upravljanje procesom se može ostvariti na različite načine, no prirodno je da se najveći broj rješenja svodi na to da upravljački sistem prima *informacije* o ponašanju procesa i da na osnovu njih i zadatog željenog ponašanja formira *komande* (*upravljanje*) kojima se mijenja ponašanje procesa.
- Da bi ovaj sistem mogao da radi neophodno je da se obezbijedi ciklično obavljanje sledećih aktivnosti: *mjerenje*, *određivanje upravljanja* i *izvršavanje komandi*. To zapravo znači da je potrebno da se omogući da računar prihvati izmjerene veličine, obradi ih na određeni način i prenese rezultate obrade na izvršne organe. U principu ove aktivnosti se obavljaju periodično, pri čemu veličina perioda zavisi od osobina procesa kojim se upravlja. Pored toga, treba omogućiti i komunikaciju između operatora i računara tokom koje će on postavljati zahtjeve u pogledu načina rada procesa i dobijati informacije o stanju procesa.

Informacije o trenutnom stanju procesa dobijaju se mjerenjem. Sve informacije su u formi električnih signala koji mogu kontinualni ili digitalni. Ovi signali se unose u računar preko posebnih procesnih ulaznih uređaja.

Upravljanje zavisi od vrste procesa. Po pravilu se definiše osnovni algoritam koji se odvija periodično (*aktivnosti inicirane protokom vremena*). Pored toga, predviđaju se i posebne upravljačke akcije koja se aktiviraju ukoliko se proces nađe u predhvarijskom stanju (*aktivnosti inicirane događajem*).

Upravljački signali koje primaju izvršni organi mogu takođe biti dvojake prirode – analogne ili digitalne. Izračunato upravljanje treba, prema tome, da se preko posebnih procesnih izlaznih uređaja transformiše u zahtijevani oblik električnog signala i prenese na izvršne organe.



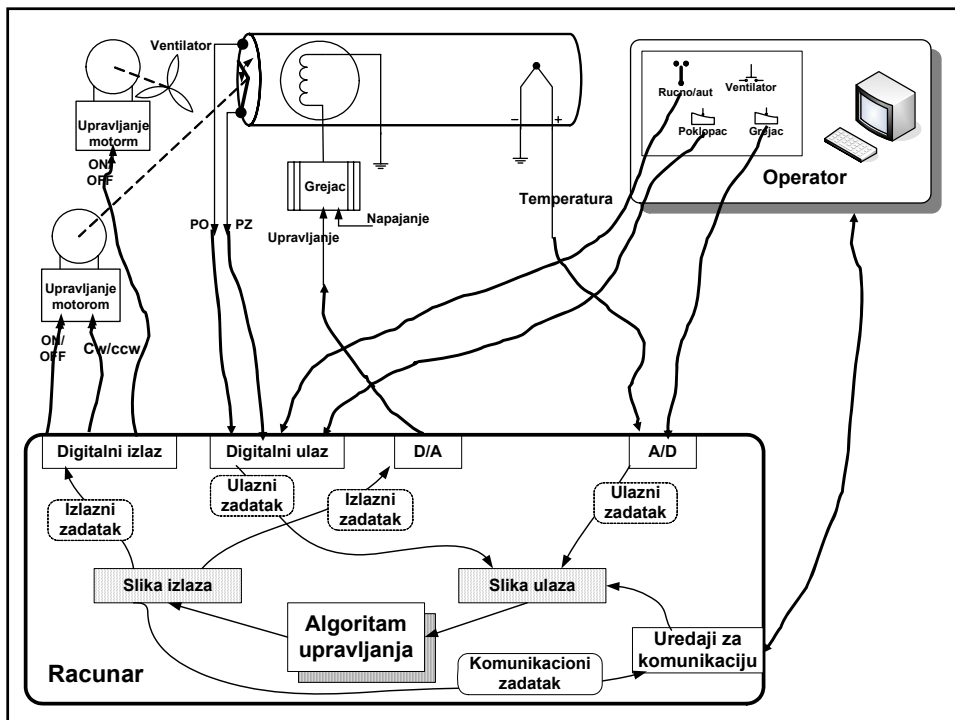
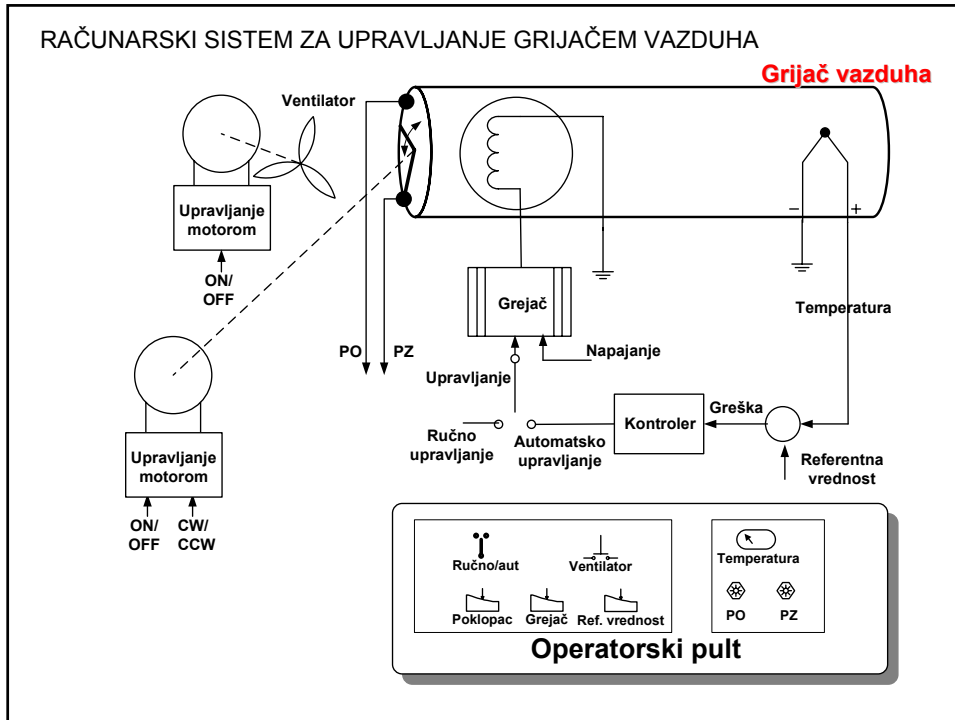
Činjenica da algoritam upravljanja (kontroler) nije direktno fizički spregnut sa procesom dovodi do čitavog niza specifičnosti pri implementaciji računarskog upravljanja u odnosu na upravljanje koje se ostvaruje pomoću analognih komponenti. Naime, kod klasičnih sistema upravljanja postoji fizička veza između kontrolera i procesa što znači da se svaki fenomen na procesu na izvestan način direktno odslikava i na ponašanje kontrolera. Kod računarske primjene, međutim kontroler operiše nad podacima za koje "vjeruje" da odgovaraju izlazima procesa i na osnovu njih izračunava upravljanje za koje "vjeruje" da se prenosi na izvršne organe. Ukoliko dođe do bilo kakvog poremećaja u ponašanju izvršnih organa kontroler to neće registrovati. On će jednostavno iz primljenih vrijednosti mjernih veličina u narednom trenutku, "videti" da je greška i dalje velika, pa će "vjerujući" da su prethodno zadana upravljanja adekvatno primijenjena na proces "zaključiti" da mora i dalje da mijenja upravljanje. Drugim riječima činjenici da ne postoji direktna fizička sprega kontrolera i procesa mora se posvetiti posebna pažnja

Komunikacioni zadaci omogućavaju vezu operatora procesa sa računarom. U tom smislu operator preko tastature može dodavati ili menjati neke informacije koje se nalaze u ulaznom području memorije ili u samom upravljačkom zadatku. Isto tako, informacije o procesu dobijene mjerenjem ili izračunate u okviru upravljačkog zadatka mogu biti, preko komunikacionog zadatka prikazane na displeju ili šampaču. Konačno svi podaci u memoriji mogu se mijenjati i na osnovu komandi koje dolaze iz računarske mreže, ili se slati drugim računarima u mreži.

Kod jednostavnih procesa komunikacija između procesa i operatora svodi se na uključivanje ili isključivanje određenih prekidača i uključivanje ili isključivanje signalnih sijalica. Ova vrsta informacija se unosi i iznosi iz računara na isti način kao i mjerni i upravljački signali. Međutim, u mnogim primjenama potreba za komunikacijom prevazilazi ove jednostavne uređaje. Inženjeri na procesu, piloti, kontrolori saobraćaja itd. zahtjevaće detaljne informacije o svim aspektima rada procesa, aviona ili saobraćajnog sistema. Otuda računar obično omogućava i komunikaciju sa standardnim ulazno-izlaznim uređajima kao što su tastatura, video displej ili štampač. Konačno, kod složenih procesa jedan računar, po pravilu, upravlja jednim dijelom procesa i nalazi se u računarskoj mreži sa drugim računarima koji upravljaju preostalim djelovima procesa. Samo se po sebi razumije da se u cilju koordinacije rada cijelog sistema mora obezbijediti mogućnost komunikacije između ovih računara u mreži.

RAČUNARSKI SISTEM ZA UPRAVLJANJE GRIJAČEM VAZDUHA

Grijač vazduha



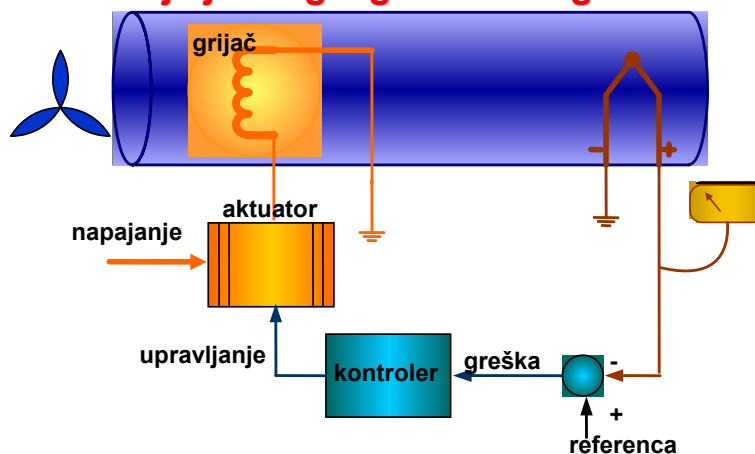
REALIZACIJA AKTIVNOSTI UNUTAR JEDNOG VREMENSKOG CIKLUSA

Svi programski zadaci koji su razvijeni u računaru moraju se odvijati sekvencijalno i to u sledećem redosledu –

- ulazni zadaci,
- upravljački zadaci,
- izlazni zadaci i
- komunikacioni zadaci.

Pri tome se ovaj niz sukcesivno ponavlja za sve vrijeme računarskog upravljanja procesom. Isto tako, neki od ovih zadataka se mogu odvijati i paralelno, pri čemu se naravno javlja problem njihove sinhronizacije.

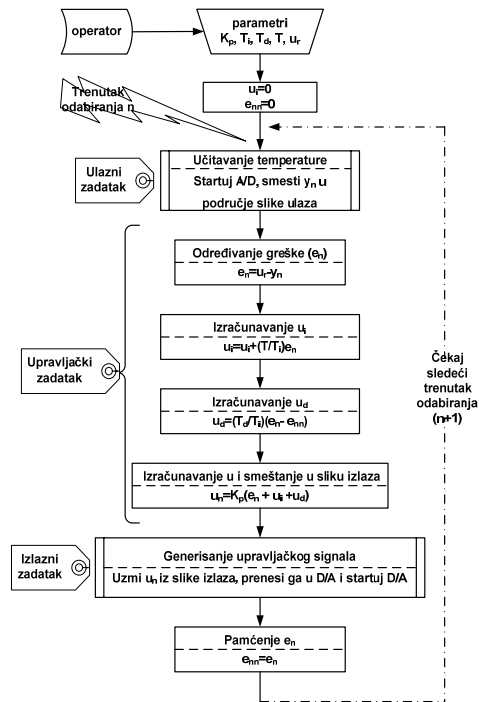
Realizacija jednog algoritamskog koraka



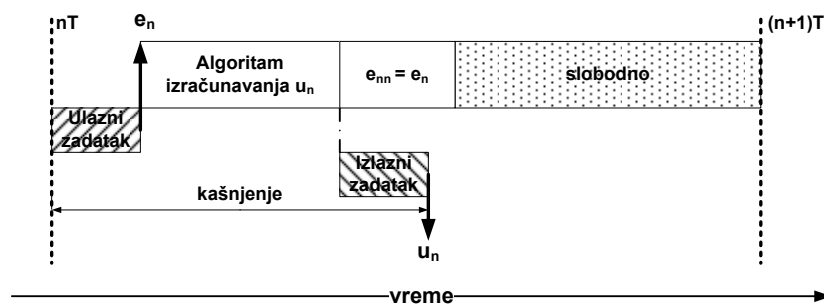
$$u(nT) = K_p e(nT) + u_i(nT) + K_p T_d \frac{e(nT) - e[(n-1)T]}{T}$$

$$u_i(nT) = u_i[(n-1)T] + \frac{K_p}{T_i} T e[(n-1)T]$$

Grafička šema jednog algoritamskog koraka

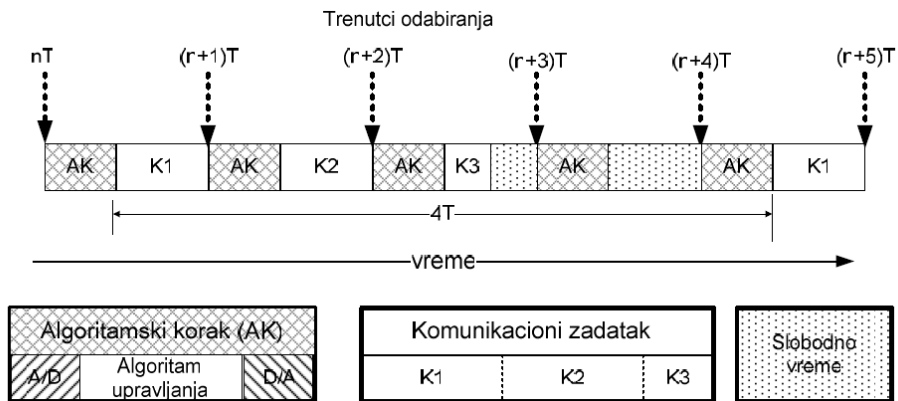


Vremenski dijagram jednog algoritamskog koraka



Kao što je već istaknuto, ulazni zadatak će biti aktiviran vremenskim događajem. Izvršavanje preostala dva zadatka uslovljeno je internim događajem – završetkom prethodnog zadatka. Potrebno je obratiti pažnju na činjenicu da će uvek postojati neko kašnjenje od trenutka odabiranja (nT) i trenutka u kome upravljački signal dolazi na aktuator

Samo se po sebi razumije da se slobodan period može iskoristiti za realizaciju komunikaciong ili nekih drugih zadataka. U principu, ovaj zadatak nema tako striktna vremenska ograničenja i njegovo izvođenje se, po pravilu, može protegnuti na više perioda odabiranja.



Projektovanje računarske aplikacije

Formiranje aplikacije najčešće otpočinje diobom svih aktivnosti na zasebne programske cjeline – zadatke. Svaki od ovih zadataka se onda zasebno programira, s tim što oni razmjenjuju podatke preko određenih područja memorije. Pri tome se, istovremeno, utvrđuje i redosled obavljanja zadataka i potreba za sinhronizacijom.

Razvijeno je nekoliko različitih apstraktnih jezika za modeliranje računarskih aplikacija. Kao i obično, svaki od ovih jezika posebno je pogodan za određene vrste aplikacija. Jezik *MASCOT* (modular approach to software construction operation and test) pokazuje se kao posebno pogodan za modeliranje računarski upravljanih sistema. Ovaj jezik koristi apstraktne objekte za formiranje preliminarnog projekta za virtuelni računar.

Aktivnost



skup operacija koju virtuelna mašina može da realizuje kao zaseban zadatak

Pri tome treba naglasiti da se u ovoj fazi projektovanja još uvijek ne vrši konačna dioba na programske zadatke koji će biti realizovani u računaru. Drugim riječima, aktivnost definiše funkcionalnu cjelinu pri čemu se u konačnoj realizaciji nekoliko aktivnosti mogu da grupišu u jedan zadatak.

U posmatranom primjeru, kao i u najvećem broju upravljanja procesom u zatvorenoj povratnoj sprezi, jasno se uočavaju sledeće aktivnosti.

- **Odabirač** – aktivnost u okviru koje se mjeri vrijeme i generiše signal prekida posle svakih T jedinica vremena. Mjerenje vremena vrši se brojanjem signala takta koji generiše časovnik realnog vremena. Naime, u zavisnosti od vrijednosti periode T određuje se broj impulsa signala takta koji odgovara zadatom vremenu odabiranja. U odabiraču se istovremeno određuje i korak vremena sa kojim se ažurira mjerač vremena u računaru.
- **Časovnik** – aktivnost u okviru koje se koji prati proteklo vrijeme u koracima koje određuje odabirač.
- **Upravljanje** – aktivnost u okviru koje se prihvataju izmjerene vrednosti sa procesa i na osnovu algoritma upravljanja izračunavaju potrebni izlazi koji se prenose na izvršni organ. Ovaj zadatak se aktivira (poziva na izvršavanje) radom odabirača.
- **Displej** je aktivnost koja obezbjeđuje prikazivanje na ekranu informacija o procesu i vremenu, kao i podataka koje definiše operater.
- **Operatorski ulaz** je aktivnost koja obezbjeđuje unošenje u računar podataka koje preko tastature definiše operater. Ovi podaci su parametri algoritma upravljanja, perioda odabiranja i slično.
- **Rukovodilac** je aktivnost koja na štampač prenprenosi informacije o procesu i vremenu, kao i o podacima koje definiše operater.

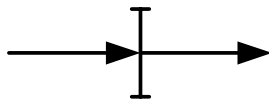
Komunikacija

Samo se po sebi razumije da različite definisane aktivnosti ne mogu da se realizuju odvojeno, jer one treba da obavljaju funkcije koje su međusobno povezane. Otuda se javlja potreba za uspostavljanjem komunikacije između aktivnosti i sinhronizacije njihovog rada.

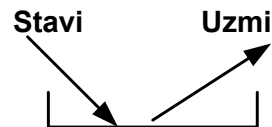
Komunikacije između aktivnosti mogu da se podijele u tri kategorije:

- direktna razmjena podataka između dvije aktivnosti
- dioba informacija između više aktivnosti
- sinhronizacioni signali

Kanal



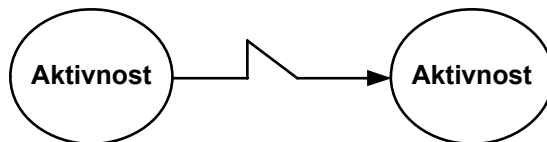
Pul (Pool)



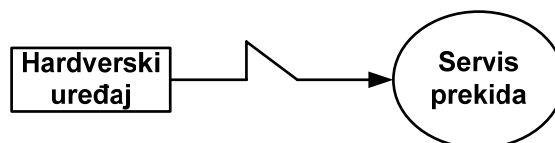
Sinhronizacija

Predstavlja se pomoću dvije procedure

- **WAIT (event)** – aktivnost se prekida (suspenduje) i čeka na događaj (event) koji je prouzrokovao suspenziju
- **SIGNAL (event)** – ova procedura "saopštava" (obznanjuje) da se odgovarajući događaj desio, sve aktivnosti koje su u stanju čekanja tog događaja nastavljaju rad



Spoljni događaj sa procesa se može posmatrati kao SIGNAL



KLASIFIKACIJA PROGRAMA

Dioba aktivnosti računarskog sistema na zadatke koji se obavljaju i ne obavljaju u realnom vremenu, kao i dalja dioba prvih na podkategorije potiče iz potrebe da se istaknu specifičnosti u projektovanju i implementaciji različitih aplikacija. Sa te tačke gledišta, u najširem smislu programi se mogu podeliti u tri kategorije:

- *sekvencijalni programi*
- *multi-tasking programi*
- *programi za rad u realnom vremenu*

Ova definicija podjele zasniva se na postupcima kojima se proverava ispravnost samog programa.

• Sekvencijalni programi

U sekvencijalnom programu aktivnosti su striktno poređane u nekom redosledu. Rezultat izvođenja programa zavisi isključivo od pojedinih aktivnosti i njihovog mjesta u programu. Vrijeme izvođenja pojedinih aktivnosti nema nikakvog uticaja na krajnji rezultat. Otuda se provjera ispravnosti ovakvih programa svodi samo na utvrđivanje da li skup pojedinih naredbi definiše određenu aktivnost i da li je struktura programa formirana tako da se proizvede željeni redosled aktivnosti. Očigledno je da se ispravnost ovakvih programa može u potpunosti provjeriti.

Multi-tasking programi

Za razliku od sekvencijalnih programa, kod multi-tasking programa pojedine aktivnosti mogu biti povezane odgovarajućim uzročno-posledičnim vezama. Ovakvi programi se sastoje od niza sekvencijalnih cjelina (zadataka) koje se izvode paralelno ili jedna drugu uslovljavaju, a koje međusobno komuniciraju preko zajedničkih promenljivih ili signala za sinhronizaciju. Ovakvi programi se provjeravaju na isti način kao i sekvencijalni, s tim što je neophodno da se u proceduru provjere uključi i sinhronizacija pojedinih zadataka. Činjenica da se pojedini zadaci mogu odvijati paralelno i da oni mogu da koriste zajedničke promenljive dovodi do nemogućnosti potpune provjere ispravnosti rada ovih programa. Da bi se to postiglo neophodno je da se uvedu dodatna pravila koja određuju hijerarhiju izvođenja konkurentnih zadataka kao i proceduru korišćenja zajedničkih promenljivih. Potrebno je ipak da se istakne da kod ovih programa samo vrijeme izvođenja pojedinog zadatka ne utiče na provjeru ispravnosti rada programa.

Programi za rad u realnom vremenu

Program za rad u realnom vremenu razlikuje se od multi-tasking programa po tome što redosled aktivnosti nije određen od strane programera već zavisi od okruženja u kome se dešavaju različiti događaji. Ovi događaji izvesno ne podijele pravilima za internu sinhronizaciju niti se može očekivati da pojedini zadatak čeka na odgovarajući signal za sinhronizaciju. Otuda je kod ovih programa vrijeme potrebno za izvođenje pojedinog zadataka od suštinskog značaja u procesu provjere ispravnosti programa.

Programiranje u realnom vremenu bilo je dugi niz godina dodatno otežano činjenicom da su programski jezici višeg nivoa bili po svojoj prirodi namijenjeni pravljenju sekvencijalnih programa i da nijesu dozvoljavali direktan pristup nizu hardverskih komponenti računara. Djelovi programa za rad u realnom vremenu veoma dugo su se posebno projektovani korišćenjem odgovarajućih assemblera. Ovaj problem je konačno prevaziđen razvojem programskog jezika "C".

Nezavisno od jezika na kome se programira programi za rad u realnom vremenu nikada ne mogu biti do kraja provereni. Ovo je direktna posledica činjenice da se ne mogu ostvariti sve moguće kombinacije spoljnih događaja koji određuju redosled aktivnosti.

Budući da ovi programi predstavljaju deo složenih tehnoloških sistema čiji neispravan rad može dovesti do katastrofalnih posledica, jasno je da se projektovanju ovakvih sistema mora posvetiti izuzetna pažnja.

Automatizacija procesa

Nivoi vođenja procesa i funkcije automatizacije

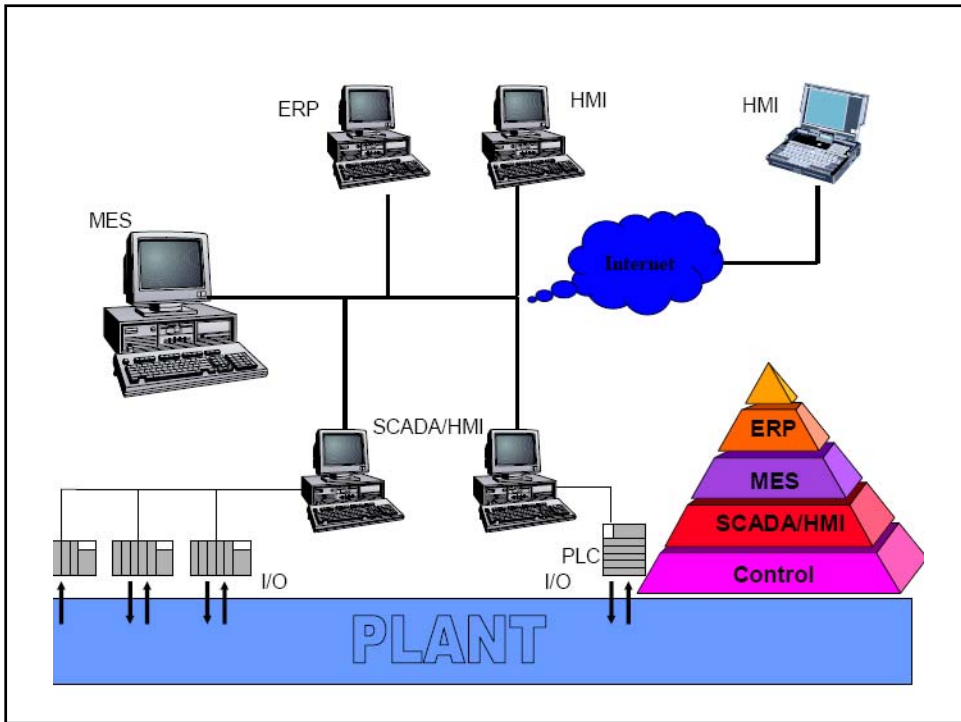
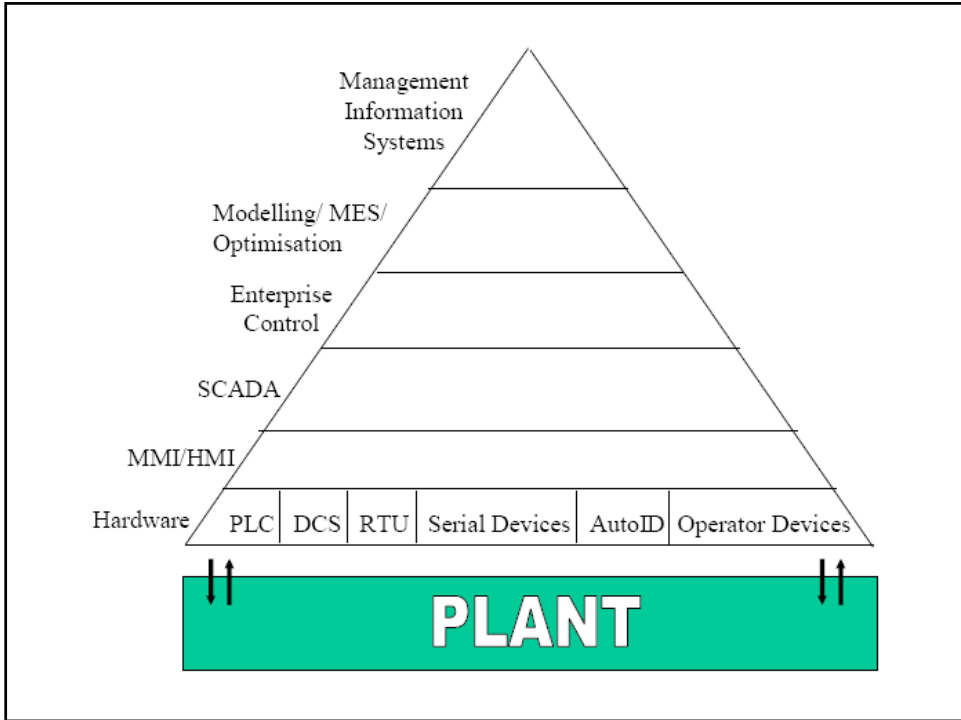
- ❖ Pri vođenju složenih tehničkih sistema koristi se princip hijerarhije između nivoa odlučivanja i nivoa izvođenja.
- ❖ Na višim nivoima vođenja pretežno se obavljaju zadaci odlučivanja i usmjeravanja, dok na nivoima "blizu" procesu prevladavaju operativni zadaci.
- ❖ Na nižim nivoima prevladavaju "sirovi" podaci. Prema višim nivoima količina podataka o procesu se smanjuje, ali se povećava specifični informacijski sadržaj.
- ❖ Suštinski je zadatak sistema za automatizaciju da se što više funkcija tehničkog procesa automatski obavlja na ekonomski prihvatljiv način.

Nivoi vođenja u složenim teh. sistemima

nivo	Naziv nivoa	ZADACI
1	Vođenje preduzeća <i>(Corporate Management Level)</i>	Vođenje preduzeća <i>(odlučivanje i usmjeravanje)</i>
2	Vođenje pogona <i>(Production Management Level)</i>	Vođenje pogona / tvornice <i>(odlučivanje i usmjeravanje)</i>
3	Vođenje postrojenja / procesa <i>(Process Management Level)</i>	Vođenje velikog tehničkog procesa <i>(operativni zadaci i dijelom zadaci odlučivanja i usmjeravanja)</i>
4	Upravljanje i regulacija <i>(Process Control Level)</i>	Sprovođenje operativnih zadataka vezanih uz regulaciju, upravljanje, nadzor i zaštitu
5	Tehnički proces	Mjerenje procesnih veličina i djelovanje na tehnički proces

Funkcije automatizacije

nivo	Naziv nivoa	FUNKCIJE AUTOMATIZACIJE
1	Vođenje preduzeća	Analiza troškova
2	Vođenje pogona <i>(Production Management Level)</i>	Planiranje toka proizvodnje, optimiziranje proizvodnih kapaciteta, obrada procesnih rezultata
3	Vođenje postrojenja / procesa <i>(Process Management Level)</i>	Nadzor nad procesom, optimiziranje, pokretanje i zaustavljanje, dijagnostika, procesna sigurnost
4	Upravljanje i regulacija <i>(Process Control Level)</i>	Prihvatanje informacija o procesnim veličinama, prihvat informacija o kvalitetu proizvoda, upravljanje, regulacija, blokiranje, funkcije u slučaju potrebe, zaštita
5	Tehnički proces	Mjerenje procesnih veličina, djelovanje na procesne veličine preko izvršnih elemenata



Klasifikacija tehničkih procesa

Podjele po industrijskim granama

- Prema vrsti medija koji se oblikuje ili transportuje:
 - materijalni procesi (npr. procesi za proizvodnju čelika, hemijski procesi),
 - energetski procesi (proizvodnja i distribucija energije),
 - informacijski procesi (prenos i obrada informacija).
- Prema načinu djelovanja na medij koji se oblikuje ili transportuje :
 - proizvodni procesi,
 - distributivni procesi,
 - procesi skladištenja.
- Obzirom na pretvaranje ili transport materijala:
 - tehnološki procesi (fizičko-hemijsko djelovanje na sirovinu),
 - proizvodnotehnički procesi (proces obrade i oblikovanja),
 - transportni procesi.

Podjela po procesnim veličinama

Za automatizaciju tehničkih procesa od primarne su važnosti procesne veličine (procesne varijable) kojima je potrebno upravljati.

- U tehničkim procesima razlikujemo tri vrste procesnih veličina:
 - Kontinualne procesne veličine (npr. temperatura u dijelu sistema),
 - Diskretne procesne veličine (najčešće binarne procesne veličine kojima se prikazuje slijed stanja pri pokretanju ili zaustavljanju postrojenja),
 - Objektu usmjerene procesne veličine koje se pridružuju pojedinim objektima. Pri tome se može raditi o fizičkim veličinama sa kontinualnim područjem vrijednosti (npr. temperatura ingota koji ulazi u valjački stan), ali i o nefizičkim veličinama (npr. tip proizvoda, skladišni broj).

Prema navedenoj podjeli procesnih veličina definišu se pripadajuće vrste procesa:

- Kontinualni procesi (*continuous processes*),
- Sekvencijalni procesi (*sequential processes, discrete event type processes*),
- Objektu usmjereni (komadni) procesi (*discrete object type processes*).

Vođenje kontinualnih procesa

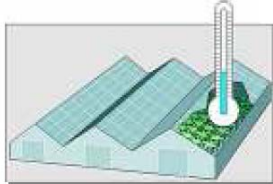
Kontinualni proces

- proces traje duže vremena bez prekidanja
- mijenjaju se samo određeni parametri ali ne i postupak
- najbolje iskorišćenje proizvodne opreme

Vođenje

- osigurava željene vrijednosti veličina
- omogućava što kraći prelaz iz jednih u druge pogonske parametre

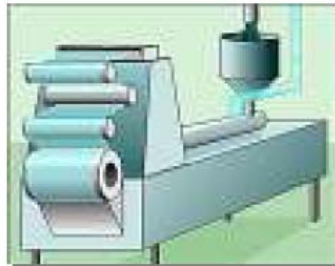
Primjeri kontinualnih procesa



Regulacija klime



Pročišćavanje otpadnih voda



Proizvodnja plastične folije



Regulacija brzine turbine u termocentrali

Vođenje šaržnih procesa

- Šaržni proces
 - slijedom operacija proizvede se određena količina proizvoda - šarža
 - takav slijed se ponavlja
 - podaci o postupku - recept
- Vođenje
 - osigurava pravilan slijed operacija
 - unutar pojedine operacije osigurava pravilne pogonske parametre

Primjeri šaržnih procesa

Priprema stočne hrane



Priprema tijesta



Priprema betona

Vođenje procesa prerade pojedinačnih proizvoda

- Proces prerade
 - slično kao kod šaržnih procesa
 - procesi u prerađivačkoj industriji i montaži
 - slijed-niz radnih operacija
 - obrada ili sastavljanje proizvoda
- Vođenje
 - osigurava pravilan slijed operacija
 - osigurava transport dijelova za obrađivanje i poluproizvoda između faza obrade

Primjeri procesa prerade pojedinačnih proizvoda

štampanje teksta na podlogu



Rezanje drvene građe

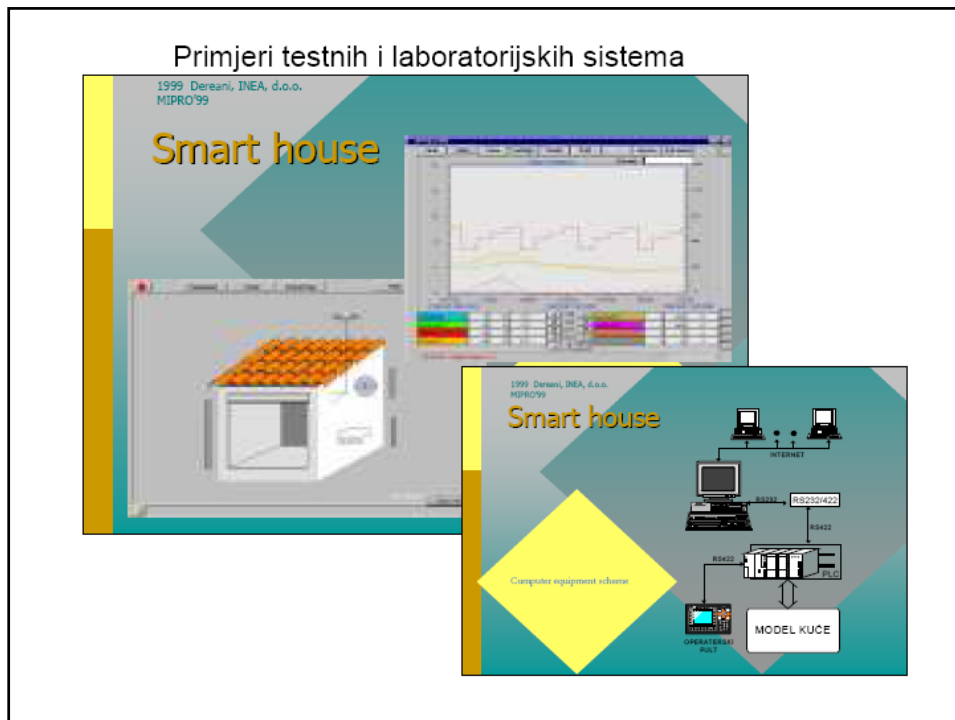


mašine za pakovanje i slaganje na palete



Testni i laboratorijski sistemi

- Umanjene varijante industrijskih procesa – obrazovanje, vježbe, eksperimenti sa novim proizvodima, testiranje kvaliteta
- Sistemi za automatizaciju laboratorijskih mjerenja i analiza
- Manje zahtjevni radni uslovi
- Upotreba personalnog računala



Ugrađeni **embedded** sistemi upravljanja

- Ugrađeni neposredno u mašine i uređaje
- Minijaturna izvedba
- Zahtjevni radni uslovi
 - temperatura, vibracije
- Veliki stepen autonomije
- Često bez veze sa drugim sistemima upravljanja
- Neposredna interakcija s operaterom tj. korisnikom

primjeri

Liftovi



Automatsko okretanje vrata

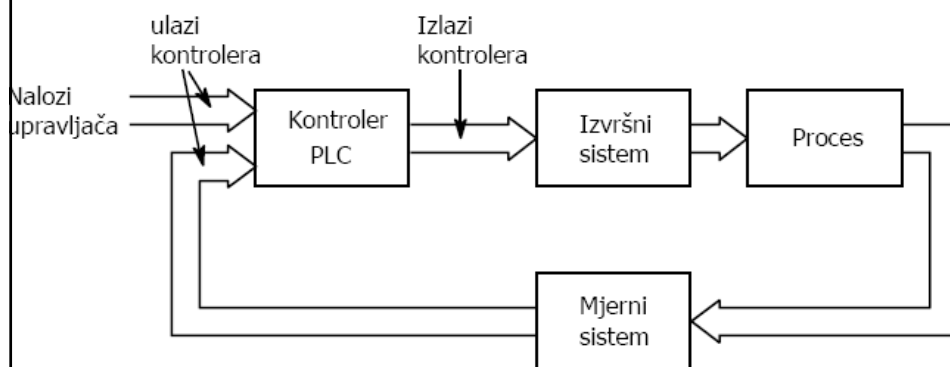
Bitna oprema u sistemima računarskog upravljanja

- Vođenje kontinualnih procesa
 - industrijski PID regulatori
- Vođenje šaržnih procesa
 - industrijski PID regulatori i programabilni logični kontroleri (PLC)
- Vođenje procesa prerade pojedinačnih proizvoda
 - programabilni logični kontroleri, roboti, CNC alati
- Testni i laboratorijski sistemi
 - personalni računari i procesni međusklopovi
- Embedded sistemi
 - mikrokontroleri, posebne izvedbe personalnih računara

Karakteristični oblici računarskog upravljanja

- Osnovno upravljanje procesa
 - logično i sekvencijsko upravljanje
 - regulacija
- Nadzorni sistemi
 - kontrola i nadzor
 - analiza podataka
 - arhiviranje podataka
 - međusklop između operatera i sistema vođenja

Logičko i sekvencijalno upravljanje

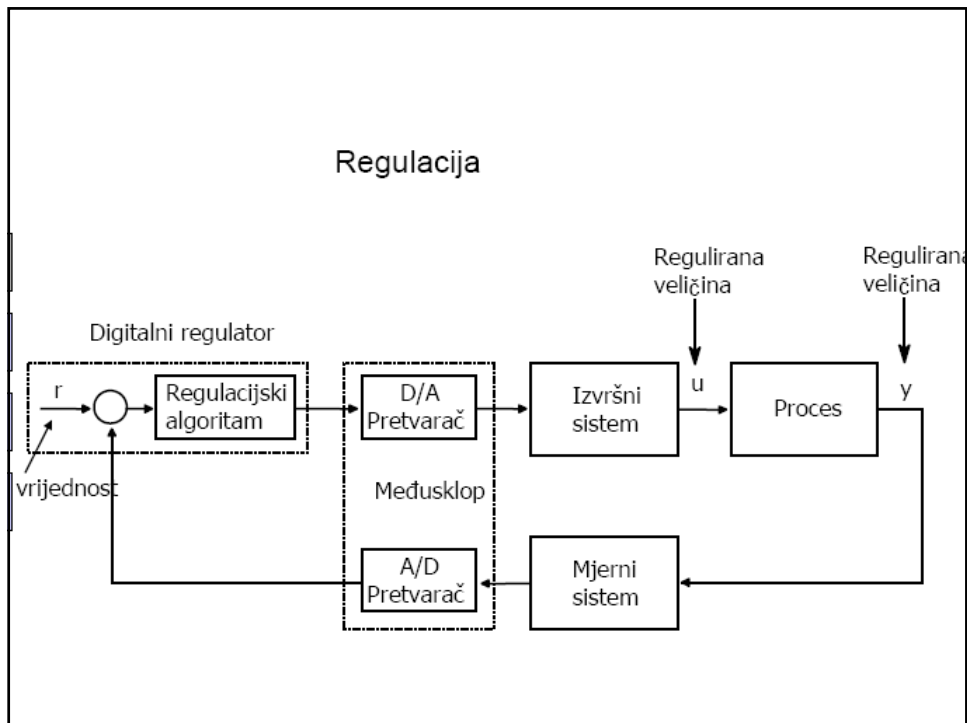


Logično i sekvencijalno upravljanje

- Mjerni sistem
 - binarni senzori i kontakti, npr.
induktivni i kapacitivni senzori blizine
fotoćelije
granični (krajnji) kontakti
- Izvršni sistem
 - binarni aktuatori i odgovarajući izvršni elementi, npr.
elektropneumatski ventili i pneumatski cilindri
releji tj. sklopnici i elektromotori
releji tj. sklopnici i električni grijači

Logično i sekvencijalno upravljanje

- **Cilj:** Izvođenje predodređenih operacija u predodređenom slijedu
- **Trajanje operacije**
 - vremenski ograničeno -> upravljanje
(do odbrojavanja vremenskog brojača u programu)
 - do nastupanja nekog događaja (npr. uključenje graničnog kontakta)
- **Primjena:** Prije svega u šaržnim i procesima prerade pojedinačnih proizvoda
- **Realizacija:** programabilni logični kontroler (PLC)



REGULACIJA

- Regulacijsko djelovanje
 - upravljanje održava regulisanu veličinu jednaku željenoj vrijednosti
- Slijedno djelovanje- praćenje
 - upravljanje osigurava da regulisana veličina, što prije i bez razlike, slijedi promjene željene vrijednosti
- Računarska realizacija regulacije
 - procesni računari
 - industrijski PID regulatori i snažniji PLC-ovi
 - važna je pouzdanost računarskog sistema

Strukture automatizacije

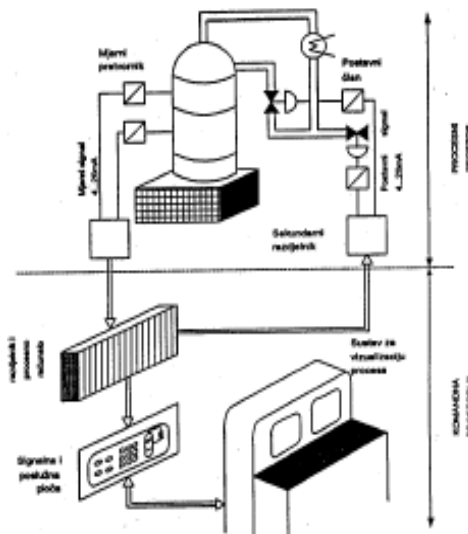
- Sistemi za automatizaciju procesa mogu se s obzirom na strukturu klasifikovati prema sljedeća tri parametra:
 - struktura tehničkog procesa,
 - razmještajna struktura opreme za automatizaciju,
 - funkcionalna struktura sistema za automatizaciju,gdje svaka od navedenih struktura može biti centralizovana i decentralizovana:

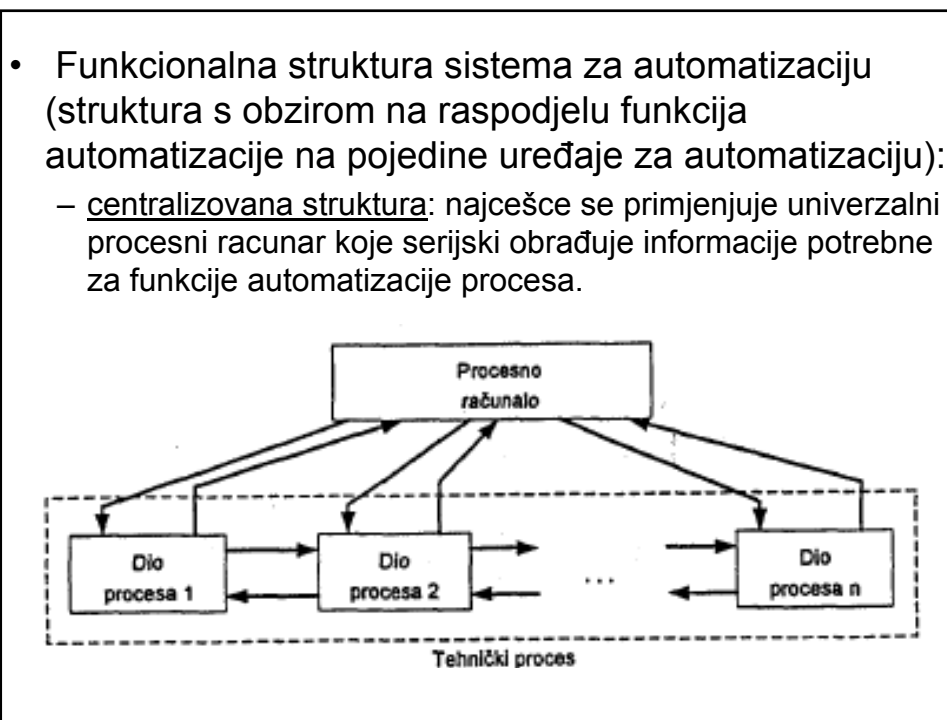
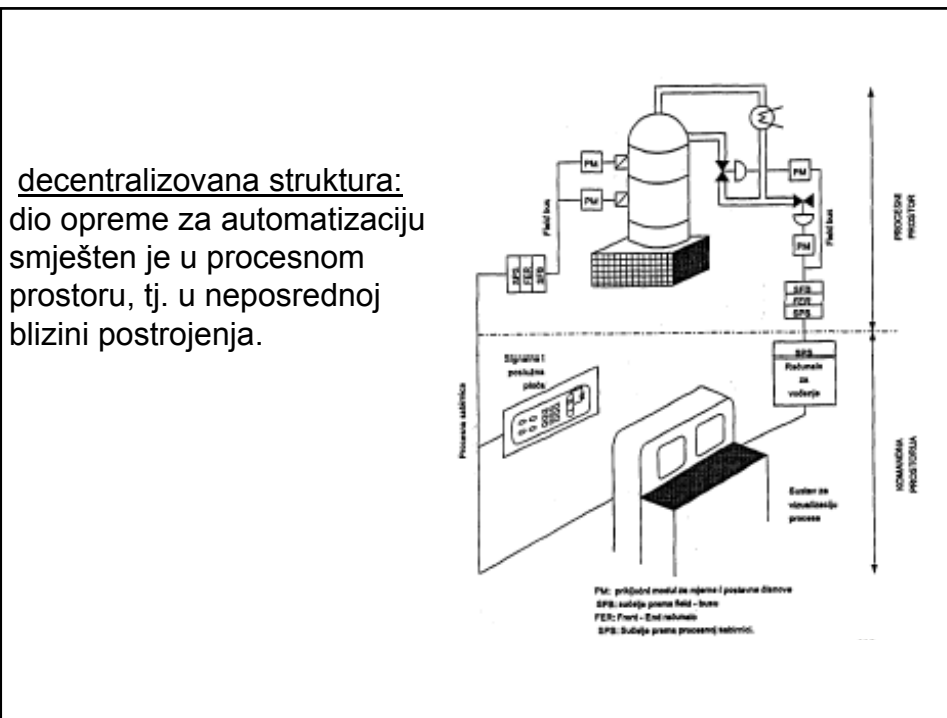
Tehnici procesi:

- centralizovana struktura: proces se može promatrati kao cjelina. Tipično se radi o manjim procesima, npr. pranje u mašini za pranje rublja,
- decentralizovana struktura: proces je moguće raščlaniti na više podprocesa.

Razmještajna struktura opreme za automatizaciju

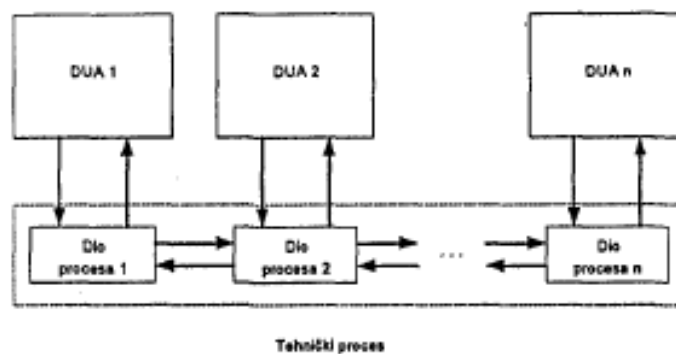
centralizovana struktura:
oprema za automatizaciju smještena je u poseban prostor – komandnu prostoriju





Funkcionalna struktura sustava za automatizaciju:

decentralizovana struktura: koriste se namjenski uredaji za automatizaciju (procesni računar) koji paralelno obavljaju pripadajuće funkcije.



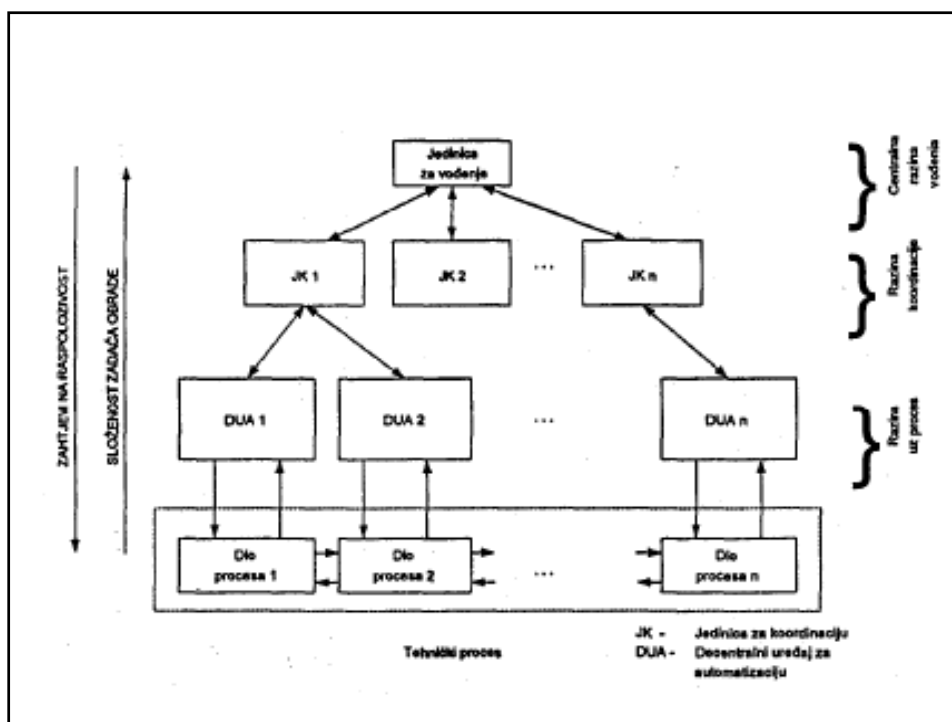
Kriterijumi za upoređivanje svojstava

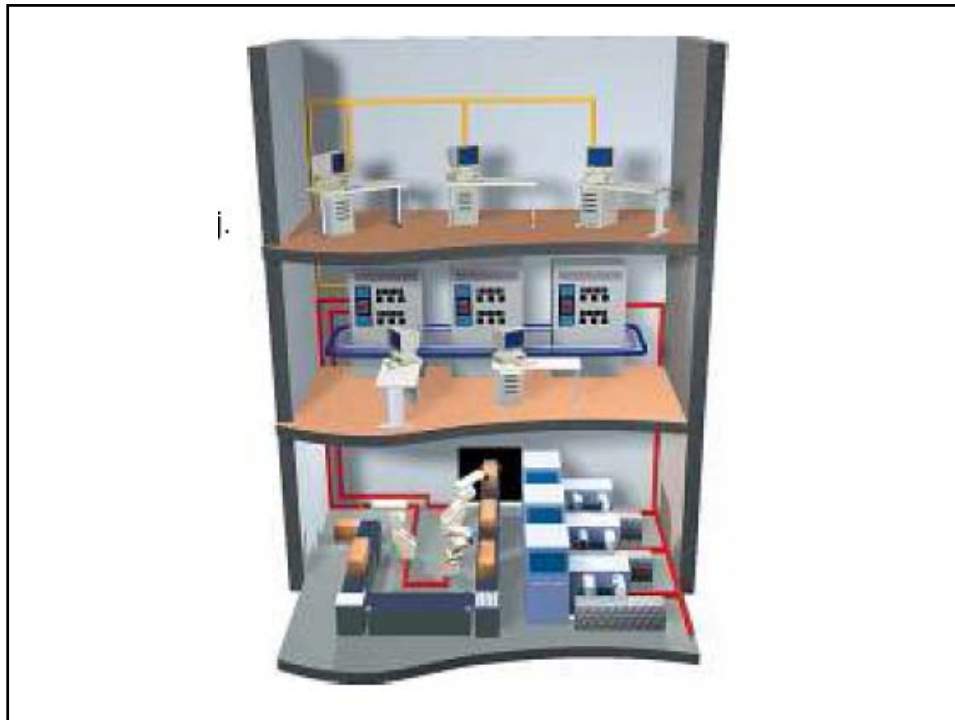
Da bi se mogle procijeniti prednosti i nedostaci strukture automatizacije razmatraju se sledeći kriterijumi:

- troškovi nabavke opreme, kabliranja, programske podrške i održavanja,
- pouzdanost i raspoloživost pri ispadu sklopova ili greškama u programskoj podršci,
- fleksibilnost pri modifikacijama,
- uskladenost dijelova procesa i optimiranje cjelokupnog procesa,
- jednostavnost rukovanja i održavanja.

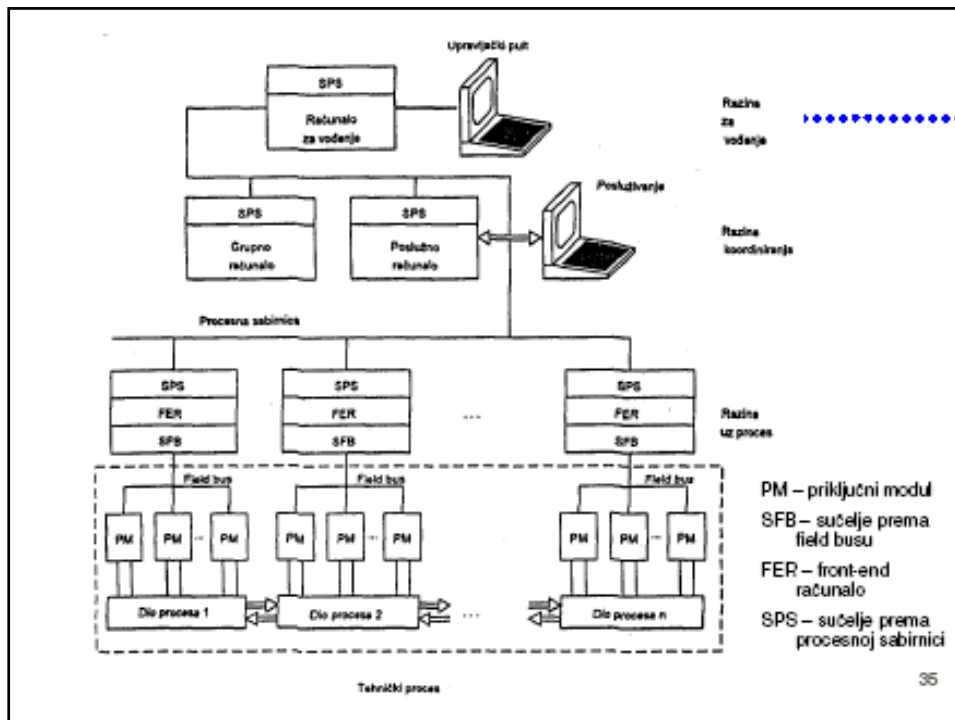
Hijerarhijska struktura automatizacije

- Hijerarhijska organizacija uređaja za automatizaciju preslikava definisane nivoe upravljana u složenim tehničkim sistemima u strukturu sistema automatizacije:
 - Decentralizovani uređaji za automatizaciju obavljaju zadatke 4. i 5. nivoa vodenja,
 - Jedinice za koordinaciju osiguravaju koordiniranje i optimiranje dijelova procesa, te nadzor i sigurnost procesa, tj. zadatke 3. nivoa,
 - Centralna jedinica za vodenje obavlja funkcije 1. i 2. nivoa.
- Zadaci obrade informacija na odredem nivou raspodijeljeni su na pojedinačne uređaje tog nivoa. Zadaci na određenom nivou uskladuje prvi nadređeni nivo hijerarhije, tako da jedinice na istom nivou međusobno ne komuniciraju.





- Hijerarhijska organizacija zahtjeva postojanje komunikacijskog sistema između pojedinih nivoa uređaja za automatizaciju.
- Dominantne su sabirničke strukture koje sadrže skup hijerarhijski organizovanih sabirnica.
- Sabirnički sistem npr. sačinjavaju:
 - "Field bus" - povezuje priključne module (procesni modul, moduli prema sensorima i izvršnim elementima) preko Front-End računara s procesnom sabirnicom,
 - Procesna sabirnica – omogućuje komunikaciju Front-End računara s uređajima na višim nivoima vodenja.



Distribuirani sistemi za automatizaciju

- Za razliku od hijerarhijske strukture u kojoj svaki uređaj komunicira samo s pridruženim uređajem višeg nivoa, kod decentralizovane distribuirane strukture decentralni uređaji za automatizaciju povezani su međusobno, kao i sa centralnim uređajem.
- U distribuiranim sistemima za automatizaciju postavljaju se sljedeći ciljevi:
 - Automatsko prilagođenje podjele zadataka između uređaja za automatizaciju,
 - Dijagnostika kvarova međusobnim nadzorom uređaja,
 - Mogućnost rekonfiguracije sistema u cilju izolacije i ograničenja kvara (*error isolation*),
 - Automatsko uspostavljanje normalnog stanja nakon otklanjanja kvara (*error recovery*).

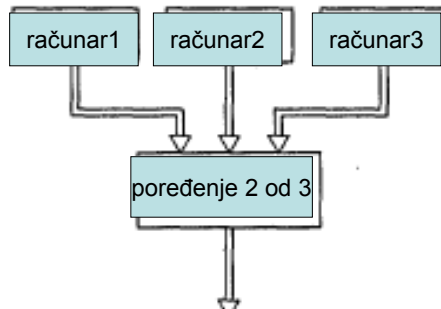
Strukture automatizacije sa redundansom

- Uvođenje redundanse u sistem za automatizaciju procesa koristi se u cilju povećanja pouzdanosti sistema. U slučaju ispada određene jedinice sistema preostale jedinice preuzimaju neke ili sve operacije koje je obavljala neispravna jedinica.
- Primjena redundanse mora osigurati da:
 - druge jedinice saznaju za ispad,
 - se sačuvaju podaci neophodni za preuzimanje operacija koje je obavljala neispravna jedinica,
 - druge jedinice preuzmu funkcije automatizacije od neispravne jedinice,
 - se ispad dojavljuje opslužnom osoblju.

- Redundansa se može realizovati kao:
 - Sklopovska redundansa (korišćenjem višestrukih sklopova),
 - Programska redundansa (korišćenjem višestruke programske podrške),
 - Mjerna redundansa (mjerenjem redundantnih, najčešće međusobno zavisnih, procesnih veličina),
 - Vremenska redundansa (ponavljanje određene operacije (mjerenja) u određenim vremenskim intervalima).

Sklopovska redundansa

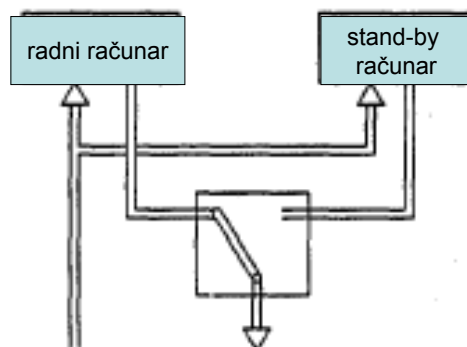
Statička ("m od n") redundansa:



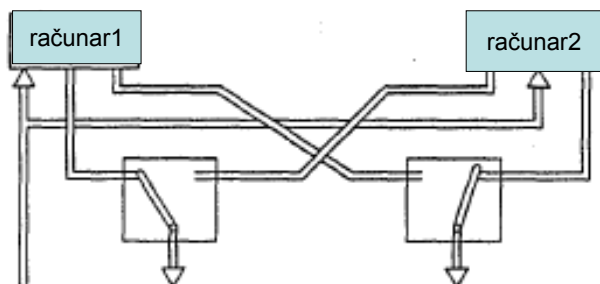
- n jedinica (računara) obavlja iste zadatke na bazi istih ulaznih podataka. Izlazni rezultati se uporeduju i na temelju većinske odluke određuje se konačan rezultat.

Dinamička - slijepa redundansa:

- Radni računar u normalnim uslovima obavlja sve funkcije automatizacije. Pomoćni (*Stand-by*) računara povezano je s radnim računarom i periodično dobija informacije o stanjima procesa, među rezultatima radnog računara i sl. U slučaju kvara radnog računara, pomoćni računar isključuje radni računar i preuzimanjegove zadatke prema unaprijed definisanoj proceduri.



Dinamička funkcionalna redundansa:



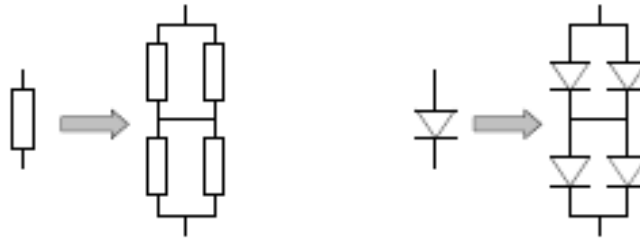
- Računar 1 obavlja zadatke neophodne za održavanje toka procesa (upravljanje procesom, zaštite), a računar 2 u normalnim uslovima obavlja manje prioritetne zadatke automatizacije (npr. optimizaciju, obrada procesnih rezultata) i nadzire računar 1.
- Ukoliko se ustanovi ispad računara 1, računar 2 preuzima njegove funkcije.

• Redundansa komunikacione strukture:

Zbog mogućnosti ispada komunikacione strukture (prekid komunikacionog medija, ispad međusklopa) i u komunikacionu strukturu potrebno je ugraditi redundansu.

- Redundansu omogućava npr. Token Ring struktura s dvostrukim prstenom.

Redundansa na nivou elektronskih elemenata:



- Prikazani spojevi osiguravaju otpornost sistema na pojavu kratkog spoja ili prekida u određenom elementu. Takva rješenja koriste se kod elemenata kritičnih za ispravan rad sistema (npr. otpornik u izlaznom kolu za upravljanje relejom koji djeluje na određenu sigurnosno kritičnu funkciju procesa).

Programska redundansa

- Programska podrška u pravilu sadrži neotkrivene greške koje se manifestuju kod pojave određene (rijetke) kombinacije ulaznih podataka.
- Programsku redundansu moguće je realizovati tako da se određeni dijelovi programske podrške izvedu na različite načine.
- To je moguće postići tako da:
 - nezavisni razvojni timovi razvijaju programsku podršku za isti modul,
 - se pri razvoju različitih verzija određenog modula primjenjuju različite strategije, algoritmi i programske strukture.

• **Izvođenje različitih programskih modula moguće je obavljati prema različitim strategijama:**

- Uzastopno izvođenje različitih verzija određenog programskog modula uz upoređenje rezultata,
- Izvođenje različitih verzija određenog programskog modula na različitim jedinicama računarskog sistema sa višestrukom redundansom,
- Cikličnim pozivanjem različitih verzija algoritma u svakoj iteraciji regulacione petlje.

