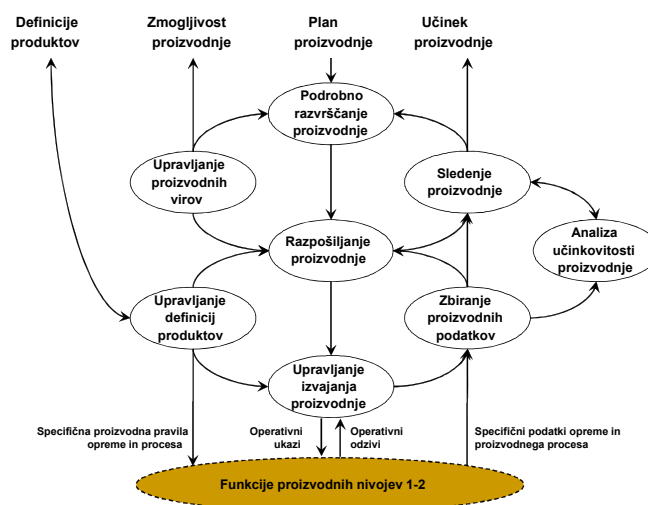


### 3. Kompleksni krmilni sistemi

- Pregled sistemov za avtomatsko vodenje proizvodnje
- Načrtovanje in izvedba
- Modularni pristopi



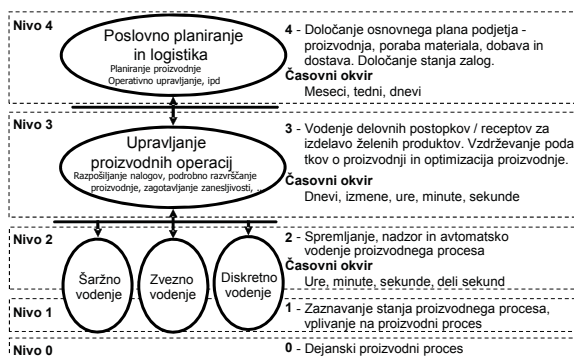
#### 3.1 Pregled sistemov za avtomatsko vodenje



## Sistemi za avtomatsko vodenje proizvodnje



- Funkcije proizvodnih nivojev 1-2

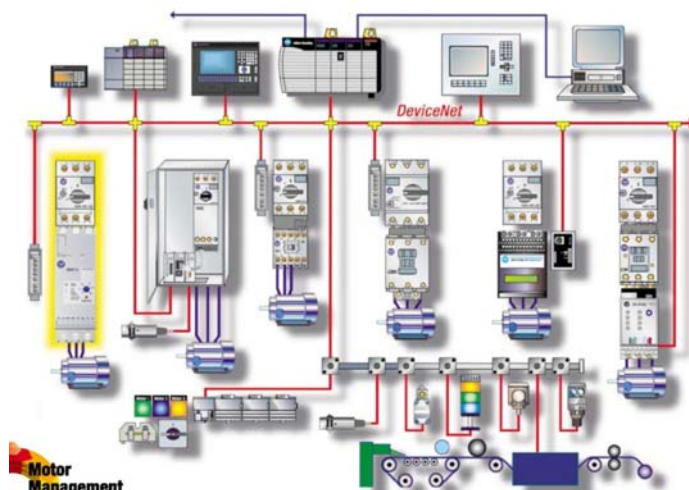


## Sistemi za avtomatsko vodenje proizvodnje - stroji



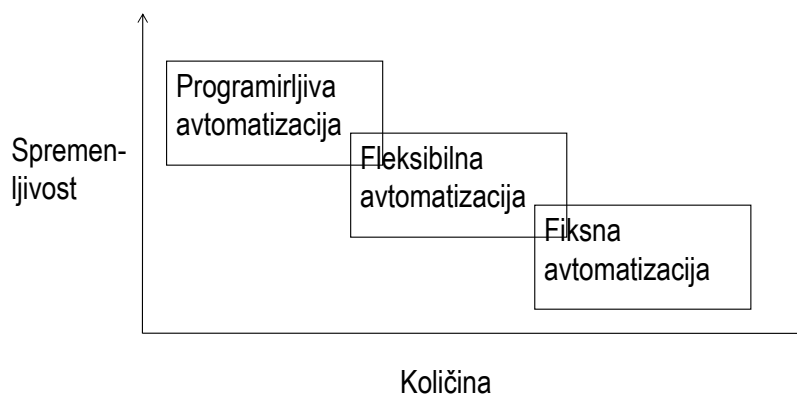
- Sistemi skladiščenja
- Sistemi rokovanja
- Sestavljalne linije
  - sestavljalne celice
  - stroji
    - aktuatorji
    - senzorji
- proizvodne linije
  - proizvodne celice
  - stroji
    - aktuatorji
    - senzorji

## Sistemi za avtomatsko vodenje proizvodnje - računalniki



- Računalniki
- Krmilniki
  - aktuatorji
  - senzorji
- Programska oprema

## Trije tipi avtomatizacije



## Fiksna avtomatizacija



- Sekvenca operacij obdelave (ali sestavljanja) je fiksna in določena s konfiguracijo opreme
- Značilnosti:
  - primerna za proizvodnjo v velikih količinah
  - velika začetna investicija v namensko načrtovano proizvodno opremo
  - visoke stopnje proizvodnje (production rates)
  - relativno nefleksibilna v prilagajanju spremembam v izdelkih

## Programirljiva avtomatizacija



- Zmožnost spreminjanja sekvence operacij preko reprogramiranja za potrebe prilagajanja različnim konfiguracijam izdelkov
- Značilnosti:
  - velika investicija v programirljivo opremo
  - nižje stopnje proizvodnje kot pri fiksni avtomatizaciji
  - fleksibilnost v soočanju z variacijami izdelkov in spremembami v konfiguraciji izdelkov
  - najprimernejša za šaržno proizvodnjo
  - med opravi oz. šaržami je potrebno spreminjati fizične nastavitve in dele programa

## Fleksibilna avtomatizacija



- Sistem je zmožen spreminjanja med enim in drugim opravilom z majhno izgubo časa med opravili
- Značilnosti:
  - velika investicija v namensko načrtovan proizvodni sistem
  - stalna proizvodnja spremenljive mešanice izdelkov
  - srednje stopnje proizvodnje
  - fleksibilnost v soočanju z „mehko“ spremenljivostjo izdelkov

## 3.2 Načrtovanje in izvedba sistemov za vodenje



- Obsega tri pomembne elemente
  - **znanja**, potrebna pri reševanju problemov vodenja (**kako?**)
    - teorija vodenja, sistemska teorija, računalništvo, informatika, elektronika, strojništvo, ekonomija, psihologija, sociologija, specifična tehnološka znanja itd.
  - **orodja** za načrtovanje in izvedbo sistemov vodenja (**s čim?**)
    - programski paketi CACSD, CASE, simulacijski paketi, ekspertni sistemi, orodja za konfiguriranje nadzornih sistemov, regulatorjev, krmilnikov itd.
  - **gradnike**, s katerimi gradimo sisteme vodenja (**iz česa?**)
    - senzorji, aktuatorji, pretvorniki, krmilniki, regulatorji, računalniški sistemi itd.

## Računalniški sistemi za avtomatsko vodenje proizvodnje



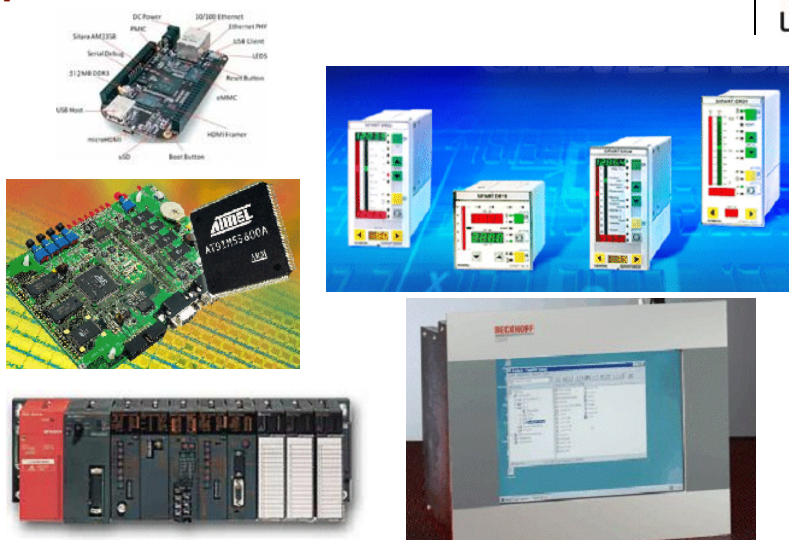
- Že pred pričetkom uporabe računalnikov so del vodenja na najnižjem nivoju izvajale namenske naprave
  - programatorji, releji
  - analogni regulatorji
- Vpeljava procesnih računalniških sistemov
  - v prvi fazi centralizirano vodenje
  - kasneje decentralizirano vodenje
  - množica mikroprocesorskih enot v omrežju
    - selitev procesne periferije od računalnika k procesu
    - digitalni prenos in obdelava signalov

## Računalniki za vodenje procesov



- Izbira odvisna od potreb in ciljev vodenja
- Osnovno vodenje procesov - namensko grajeni mikroračunalniki
  - mikrokrmilniki (mikrokontrolerji) –  $\mu$ C
  - programirljivi logični krmilniki – PLC, PAC
  - industrijski zančni regulatorji
  - procesni računalniški sistemi – DCS
  - industrijski računalniki
- Višji nivoji vodenja - splošnonamenski računalniki
  - specializirana programska oprema

## Računalniki za vodenje procesov - primeri



## Programirljivi logični krmilniki



- Najpogostejši način implementacije logičnega in sekvenčnega vodenja
  - PLC - Programmable Logic Controllers
  - SPS - Speicherprogrammierbare Steuerungen
- Definicija
 

**Programirljivi logični krmilnik (PLK)** je digitalno delujoča elektronska naprava, ki na podlagi ukazov, shranjenih v programirljivem pomnilniku, izvaja logične, sekvenčne, časovne in aritmetične operacije ter s tem vodi različne naprave in procese preko binarnih (digitalnih) in analognih vhodov in izhodov

## Mikroprocesorski PLK



- Posebna oblika procesnih računalnikov
  - prirejeni za logično in sekvenčno vodenje
  - preprosta vgradnja in priključitev signalov
  - enostavni za programiranje
  - programske operacije namesto relejskih kontaktov -> večja zanesljivost
  - programirne naprave (PC) avtomatsko generirajo dokumentacijo
  - hitro in enostavno vnašanje sprememb
  - razširjen nabor funkcij, npr. kompleksne aritmetične operacije
  - možnost komunikacije s drugimi napravami

## Zgradba in lastnosti PLK



- Mikroračunalniki, prirejeni za delovanje v industrijskem okolju
  - električne motnje, mehanski udarci in tresljaji, temperaturna nihanja, prah, vlažnost, korozivna atmosfera
- Osnovne enote
  - napajalnik
  - centralna procesna enota (CPE)
  - vhodno-izhodni moduli
- Dve izvedbi
  - kompaktna
  - modularna



## Primeri

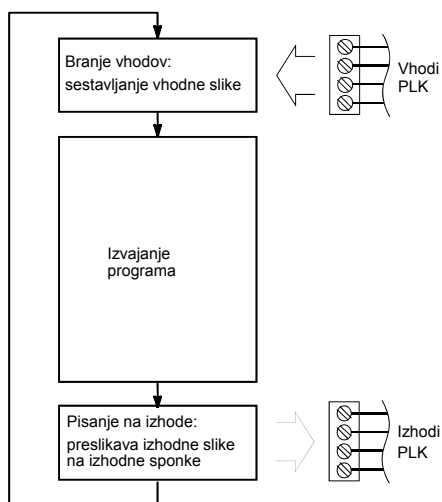


## Vhodno/izhodni moduli



- Enostavni krmilniki
  - le binarni (digitalni) vhodi in izhodi
- Zmogljivejši krmilniki - širok izbor različnih modulov
  - binarni in analogni vhodi in izhodi
  - komunikacijski moduli
  - posebni funkcijski moduli
- Zaščita
  - galvanska ločitev, filtriranje odvečnih preklpov, zaščita pred napetostnimi sunki (absorbcija konic)
  - zunanje napajanje senzorjev (vh. stikal) in bremen, pri relejskih še dodatno napajanje relejev

## Način delovanja PLK



- **Branje vhodov**
  - stanje vhodov se preslika v pomnilnik
  - »vhodna slika«
- **Izvajanje programa**
  - izračun programiranih logičnih izrazov
  - rezultati se zapisujejo v »izhodno sliko«
- **Pisanje na izhode**
  - izhodi se postavijo v skladu z vsebino pomnilnika

## Programiranje logičnih krmilnikov



- Tehnike načrtovanja logičnega in sekvenčnega vodenja so povezane z načinom izvedbe
- Načrtovanje relejskih vezij:
  - prilagojena oblika električnih vezalnih shem
- Načrtovanje krmilij v obliki logičnih vezij:
  - metode sinteze digitalnih vezij
- Načrtovanje vodenja s PLK - programiranje:
  - kombinacija uveljavljenih pristopov
  - dodatni programski jeziki

## Programski jeziki za programiranje logičnih krmilnikov



- Osnova - programski jezik, podoben zbirniku
- Ukazi za branje vhodov, logične operacije, pisanje na izhode
- Programirne naprave
  - specializirane naprave
  - osebni računalniki
- Poleg tekstovnih tudi grafični programski jeziki
  - preklopne funkcije vnašamo preko ekvivalentnih logičnih vezij ali električnih vezalnih shem
  - programirna naprava prevede program v obliko, primerno za izvajanje

## Standardizacija programiranja



- Večina PLK podpira programiranje v več programskih jezikih
- Nekaj uveljavljenih jezikov, vendar v več različicah
- Razlike med PLK različnih proizvajalcev
- Težavno programiranje različnih PLK
- Težaven prenos programov med različnimi tipi
- Potreba po poenotenju
  - mednarodni standard

## Primer razlik v programiranju



Zahteva: Izhod št. 0 naj se vključi, če je prižgan vhod št. 1 ali vhod št. 2, pri tem pa ne sme biti prižgan vhod št. 0. Če pogoj za vključitev ni izpolnjen, mora biti izhod izključen (v stanju 0)

Formalen zapis:  $y_0 = (x_1 + x_2) \cdot \bar{x}_0$

Izvedba:

SIEMENS:	MITSUBISHI:	IEC IL:
A(		
O 10.1	LD X1	LD %IX1
O 10.2	OR X2	OR %IX2
)	ANI X0	ANDN %IX0
AN 10.0	OUT Y10	ST %QX0
= Q0.0		

## Standard IEC 61131



- IEC - International Electrotechnical Commission
- Standard IEC 61131 združuje številna določila v zvezi s PLK
- Namen: poenotenje delovanja, uporabe, programiranja PLK in pripadajočega dokumentiranja
- Prenosljivost programov
- Povezljivost v komunikacijska omrežja

## Deli standarda IEC 61131 (1-2)



- IEC 61131-1 (sprejet 1992, 2. verzija 2003)
  - splošni del, terminologija, definicije pojmov, ki se uporabljajo v ostalih delih standarda
  - splošne informacije o lastnostih in uporabi PLK
- IEC 61131-2 (sprejet 1992, 2. verzija 2003, 3. verzija 2007)
  - definira zahteve za strojno opremo
  - električne, mehanske in funkcionalne zahteve
  - pogoji servisiranja, skladiščenja in transporta
  - postopki za preverjanje skladnosti s standardom

## Deli standarda IEC 61131 (3-5)



- IEC 61131-3 (sprejet 1993, 2. verzija 2003, 3. verzija 2013)
  - programiranje, struktura programa, programski jeziki, sintaksa in semantika
- IEC 61131-4 (sprejet 1995, 2. verzija 2004)
  - uporabniške smernice, analiza in specifikacija zahtev, izbira, izvedba in vzdrževanje sistemov s PLK
- IEC 61131-5 (sprejet 2000)
  - komunikacije, definira komunikacijske funkcijske bloke, ki jih lahko uporabljamo pri programiranju PLK

## Dodatni deli standarda



- IEC 61131-6 (sprejet 2012)
  - prvotno: komunikacija med PLK preko področnih vodil (opuščeno -> 61158)
  - po novem: varnostne funkcionalne zahteve za PLK
- IEC 61131-7 (sprejet 2000)
  - programiranje PLK z uporabo mehke logike
- IEC 61131-8 (sprejet 2000, 2. verzija 2003)
  - smernice za uporabo in implementacijo programskih jezikov, ki jih definira 3. del standarda
- IEC 61131-9 (sprejet 2013)
  - IO-Link – single-drop digital communication interface for small sensors and actuators (SDCI)

## IEC 61131-3 mednarodni standard za programiranje PLK



- Dva vsebinska sklopa
  - skupna programska osnova
  - standardizirani programski jeziki
- Skupna programska osnova
  - struktura programa
    - konfiguracija, viri in opravila
    - programske organizacijske enote oz. programski moduli (Program Organization Units - POU)
      - programi, funkcijski bloki, funkcije
  - podatkovni tipi in spremenljivke
  - standardne funkcije in funkcijski bloki
  - vsi elementi, ki so skupni vsem programom, ne glede na uporabljen programski jezik

## IEC 61131-3

### mednarodni standard za programiranje PLK



- Standardizirani programski jeziki
  - štirje osnovni programski jeziki
    - dva tekstovna jezika:
      - IL** seznam ukazov (Instruction List)
      - ST** strukturiran tekst (Structured Text)
    - dva grafična jezika:
      - LD** lestvični diagram (Ladder Diagram)
      - FBD** funkcijski blokovni diagram (Function Block Diagram)
  - dodaten programski jezik, namenjen strukturiranju opravil in programskih modulov:
    - SFC** sekvenčni funkcijski diagram (Sequential Function Chart)

## Standardizirani programski jeziki



- Seznam ukazov (IL)
  - osnovni programski jezik, primer:
 

<b>LD</b>	<b>%IX1</b>
<b>OR</b>	<b>%IX2</b>
<b>ANDN</b>	<b>%IX0</b>
<b>ST</b>	<b>%QX0</b>
  - omogoča uporabo oznak, skokov in komentarjev:
 

<b>LD</b>	<b>R1</b>	<b>(* Nalozi vrednost R1 *)</b>
<b>JMPC</b>	<b>RESET</b>	<b>(* Skoci, ce 'True' *)</b>
<b>LD</b>	<b>PRESS_1</b>	
<b>ST</b>	<b>MAX_PRESS</b>	
<b>RESET:</b>	<b>LD</b>	<b>0</b>
	<b>ST</b>	<b>A_X43</b>

## Standardizirani programski jeziki /2



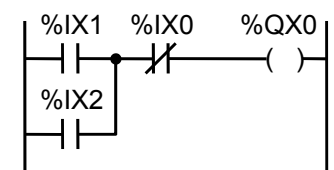
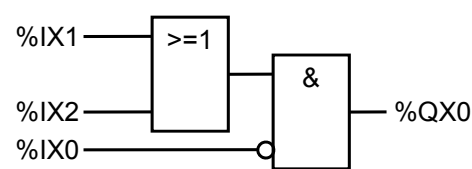
- Strukturiran tekst (ST)
  - višjenivojski programski jezik
  - sintaksa podobna Pascalu
  - prirejanje sestavljenih izrazov, primer:
 

```
%QX0 := (%IX1 OR %IX2) AND NOT %IX0;
```
  - sestavljeni stavki
 

```
IF ... THEN ... ELSE ... END_IF;
FOR ... DO ... END_FOR;
WHILE ... DO ... END_WHILE;
REPEAT ... UNTIL ... END_REPEAT;
```

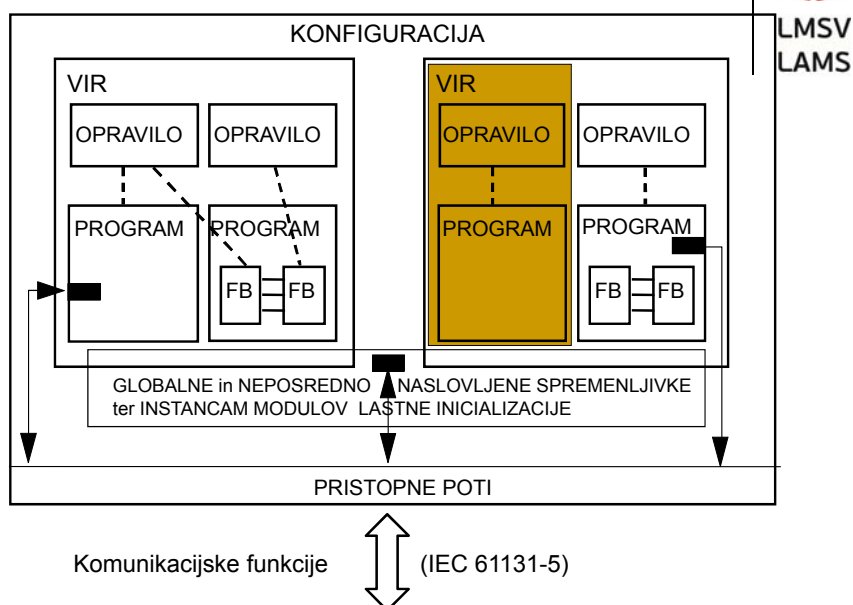
## Standardizirani programski jeziki /3



- Lestvični diagram (LD)
  - izhaja iz relejske tehnike
  - primer: 
- Funkcijski blokovni diagram (FBD)
  - izhaja iz simbolov digitalne tehnike
  - primer: 
- Sekvenčni funkcijski diagram (SFC)
  - predstavlja stanja sistema in prehode med njimi
  - običajno se uporablja v povezavi z ostalimi jeziki



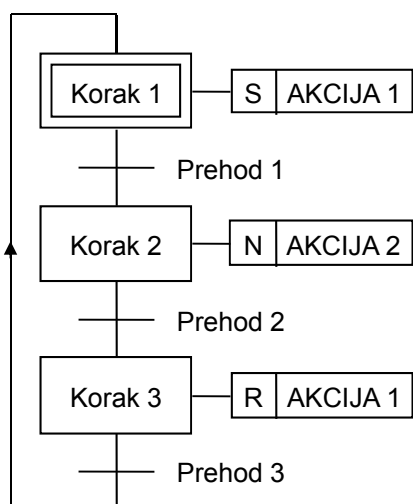
## Struktura programa



## Sekvenčni funkcijski diagram

- Definicija se naslanja na standard IEC 60848
  - standard za specifikacijo in dokumentiranje postopkov v sistemih vodenja
  - izhaja iz francoskega Grafset-a
  - osnova je teorija Petrijevih mrež
- Namen
  - strukturiranje programskih modulov, napisanih v različnih jezikih, za potrebe sekvenčnega vodenja
- Osnovni elementi
  - koraki, prehodi, akcije, prehodni pogoji

## Sekvenčni funkcijski diagram /2

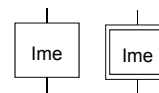


- Koraki so povezani s stanji sistema
- Začetni korak
- Korakom so pridružene akcije
- Koraku vedno sledi prehod
- Prehodom so pridruženi prehodni pogoji

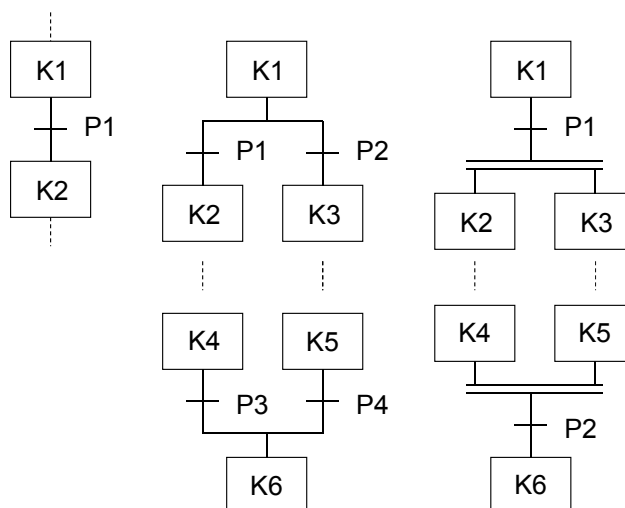
## Koraki in prehodi



- Koraki
  - korak je lahko aktiven (active) ali neaktiven (idle)
  - aktivnih korakov je lahko več
  - ob začetku izvajanja programskega modula je aktiven **začetni korak**; vsak diagram ima lahko le en začetni korak
  - grafična predstavitev – pravokotnik, povezave vstopajo vanj zgoraj in izstopajo spodaj
- Prehodi
  - uravnavajo prehajanje stanj – pričetek in prenehanje aktivnosti tistih korakov, ki so s prehodom povezani
  - vsakemu prehodu je pridružen **prehodni pogoj** – dovzetnost
  - prehodni pogoj je lahko napisan v poljubnem jeziku, rezultat mora biti dvojiška vrednost



## Povezave v diagramu



## Akcije

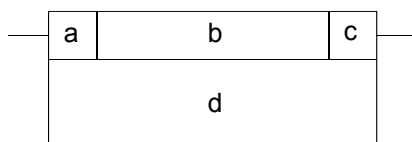


- Koraku je lahko pridružena ena, nobena ali več akcij
- Korak brez akcije predstavlja čakanje na izpolnitev prehodnega pogoja
- Akcija je lahko
  - dvojiška spremenljivka
  - del programa v poljubnem jeziku - podprogram, ki se izvaja glede na aktivnost koraka
  - sekvenčni funkcijski diagram
- Standardni SFC dopušča več načinov deklaracije akcij

## Akcije /2



- Grafična predstavitev



a - kvalifikator (določa tip akcije)  
 b - ime akcije  
 c - spremenljivka za signalizacijo  
 d - akcija v enem od jezikov  
 za programiranje PLK

- Kvalifikatorji

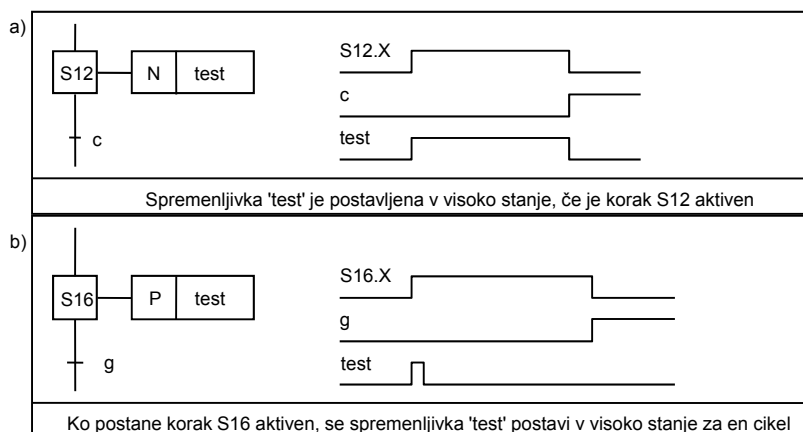
- tip akcije pove, kako je izvajanje akcije povezano z aktivnostjo pripadajočega koraka
- nekateri tipi akcij potrebujejo poleg kvalifikatorja še dodatni časovni parameter

## Tipi akcij

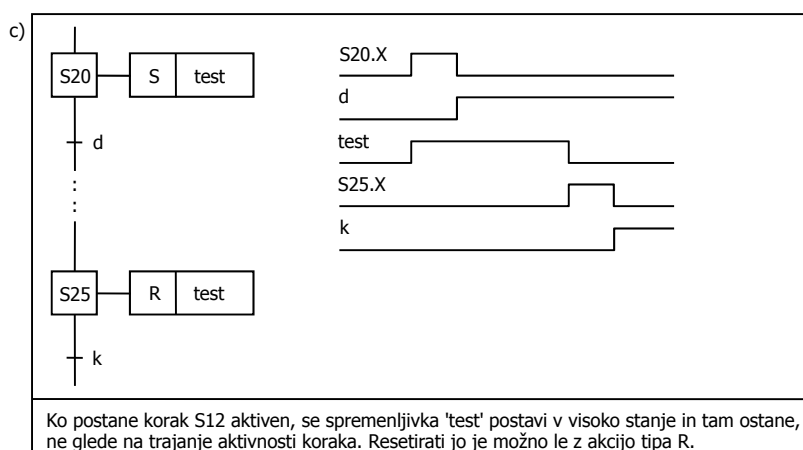


Št.	Kvalifikator	Razlaga
1	None	Non-stored (null qualifier)
2	<b>N</b>	Non-stored
3	<b>R</b>	overriding Reset
4	<b>S</b>	Set (Stored)
5	<b>L</b>	time Limited
6	<b>D</b>	time Delayed
7	<b>P</b>	Pulse
8	<b>SD</b>	Stored and time Delayed
9	<b>DS</b>	Delayed and Stored
10	<b>SL</b>	Stored and time Limited
11	<b>P1</b>	Pulse (rising edge)
12	<b>P0</b>	Pulse (falling edge)

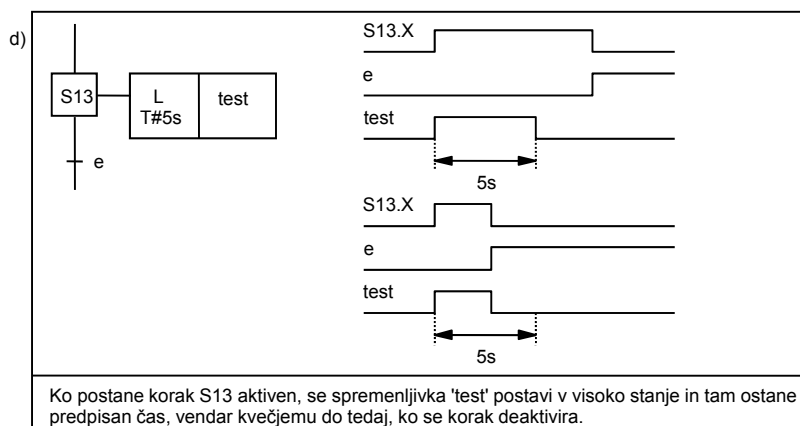
## Trajanje akcij



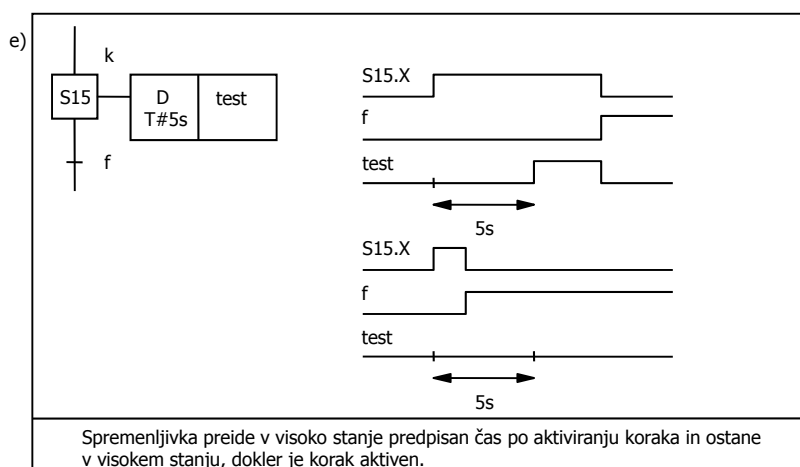
## Trajanje akcij /2



## Trajanje akcij /3



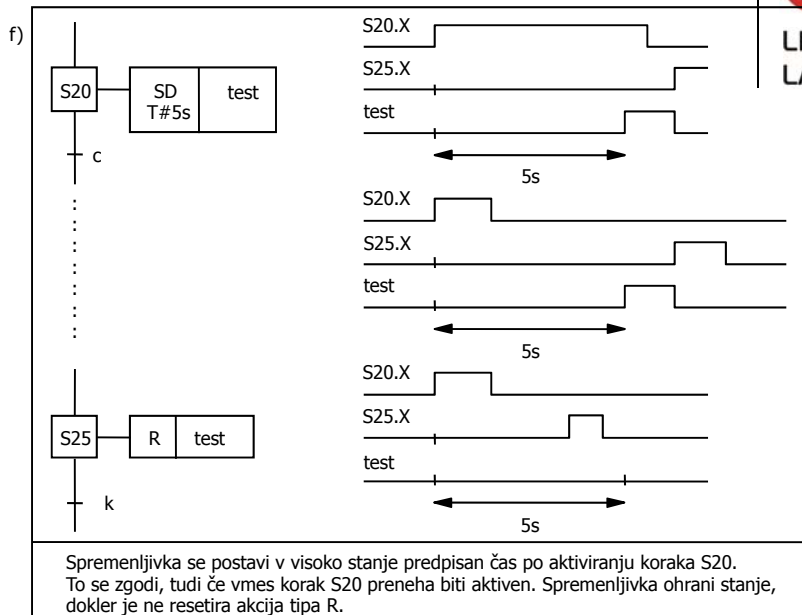
## Trajanje akcij /4



## Trajanje akcij /5



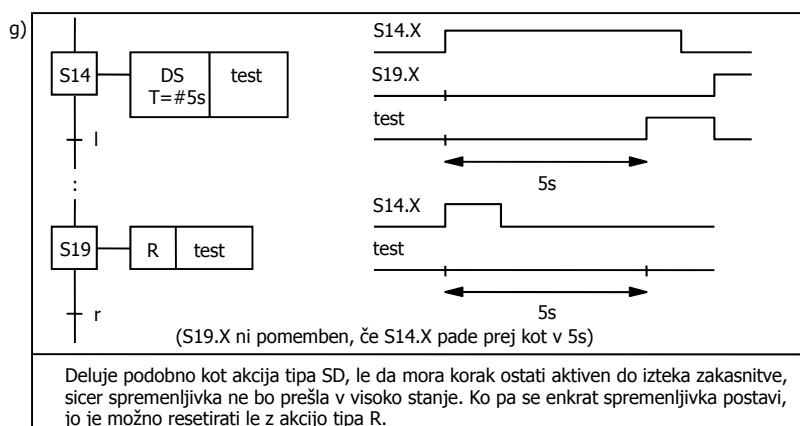
LMSV  
LAMS



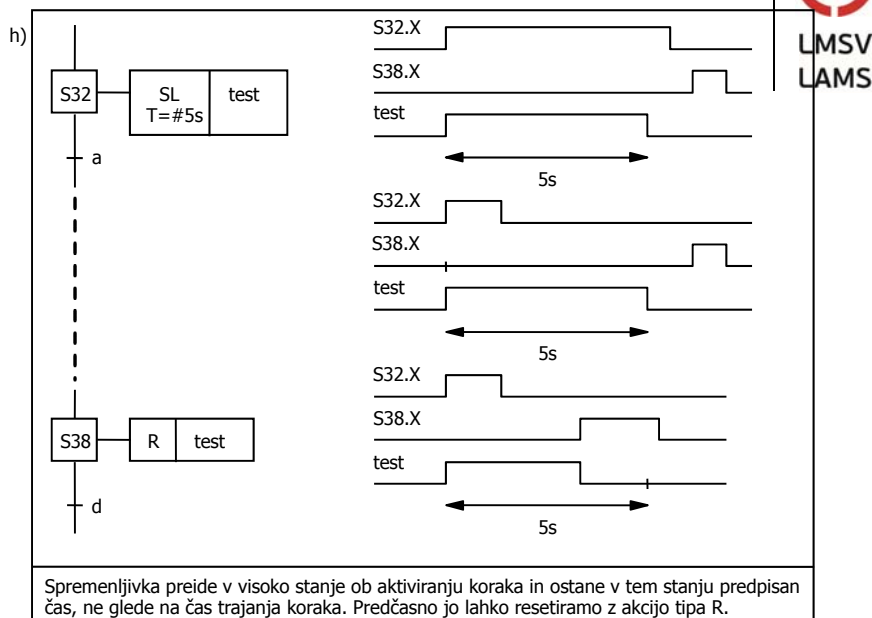
## Trajanje akcij /6



LMSV  
LAMS



## Trajanje akcij /7



## Trajanje akcij /8



- V splošnem je akcija več kot le postavljanje spremenljivke, običajno je podprogram
- Podprogram se izvaja takrat, ko bi bila akciji prirejena spremenljivka v visokem stanju
  - po 'koncu izvajanja', se podprogram v vsakem primeru izvede ŠE ENKRAT!
  - pomembno za resetiranje časovnikov ipd.
  - vsaka, tudi pulzna akcija, se torej izvede vsaj dvakrat!

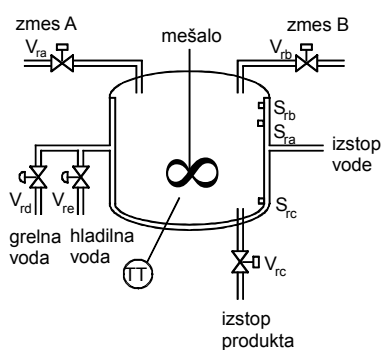




## Izvajanje

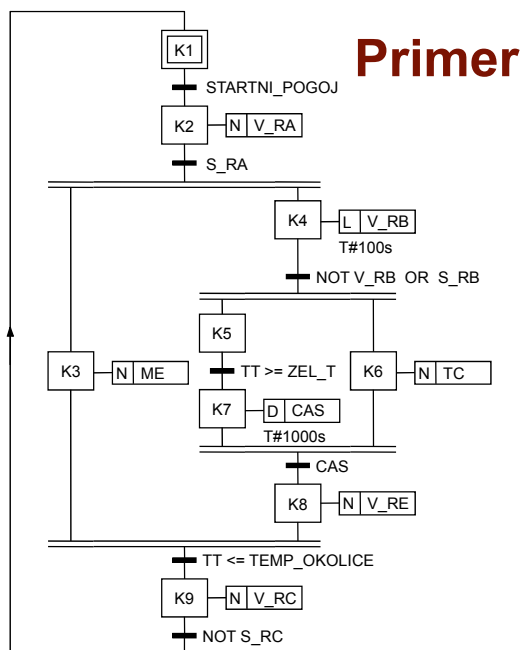
- Začetno stanje je določeno z začetnim korakom
- Sprememba stanja korakov se zgodi ob sprožitvi prehoda
- Prehod se lahko sproži, če
  - je omogočen ⇔ aktivni so vsi koraki, od katerih gredo povezave na ta prehod
  - in dovzeten ⇔ prehodni pogoj je izpolnjen
- Ob sprožitvi prehoda postanejo vsi koraki pred njim neaktivni, vsi koraki za njim pa aktivni
- Vsi prehodi, ki se v danem trenutku lahko sprožijo, se sprožijo sočasno

## Primer

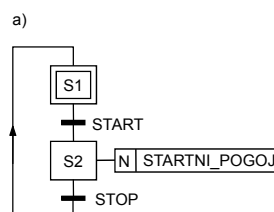


Tipki:  
 Start - pričetek obratovanja  
 Stop - zaustavitev po koncu šarže

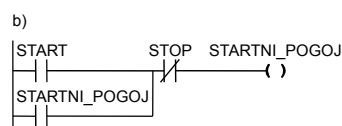
- Reaktor v šaržnem sistemu
- Recept
  - napolnimo surovino A (do stikala  $S_{ra}$ )
  - mešamo in napolnimo surovino B (100 s ali do  $S_{rb}$ )
  - vključimo regulacijo temperature
  - čakamo 1000 s pri željeni temperaturi
  - hladimo vsebino
  - izpraznimo reaktor



Za zagon/zaustavitev potreben  
vzporedno delujoč program v SFC  
ali LD:



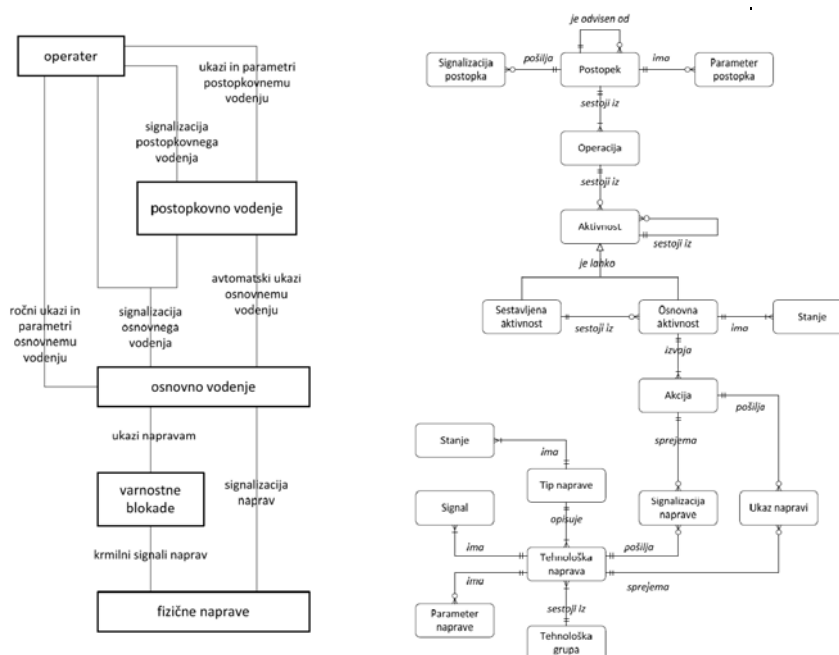
(\* STARTNI\_POGOJ je globalna spremenljivka \*)



## 3.3 Modularni pristopi



- Poskus obvladovanja kompleksnosti sodobnih avtomatiziranih sistemov
  - dekompozicija na komponente
- Strategije „modeliranja“
  - v smislu strategije razgradnje sistema na sestavne dele
- Strategije načrtovanja in izvedbe vodenja
  - modularnost
  - povezljivost
  - generiranje kode
- Preplet modeliranja, načrtovanja in izvedbe

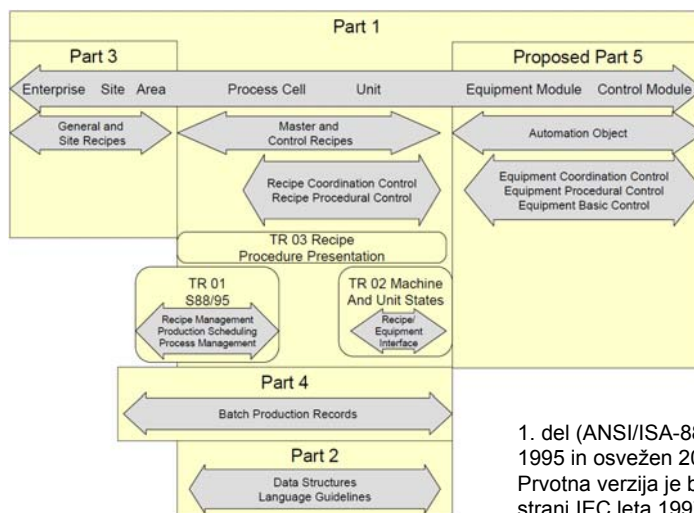


## Standard ANSI/ISA-88



- Izrazita potreba po modularnih pristopih na področju vodenja šaržnih sistemov
  - primer programirljive avtomatizacije
  - zgolj enostavnost preprogramiranja PLK ni dovolj
  - uporabnost SFC, a potreba po „kontroliranih“ pogojih spreminjanja delovanja sistema
- Standard ANSI/ISA-88, imenovan tudi S88
  - standardni model and terminologija, ki poudarja dobre prakse pri načrtovanju in upravljanju šaržnih proizvodnih procesov
  - uporablja se za izboljšanje vodenja šaržnih procesov
  - možno ga je uporabiti ne glede na stopnjo avtomatizacije procesa ter tudi za drugačne vrste procesov

## Standard ANSI/ISA-88



1. del (ANSI/ISA-88.01) je bil sprejet 1995 in osvežen 2010; Prvotna verzija je bila potrjena s strani IEC leta 1997 kot IEC 61512-1

## Šarže in šaržni procesi



- Šarža
  - material, ki se proizvaja ali je bil proizveden z enim tekom šaržnega procesa
  - entiteta, ki predstavlja proizvodnjo materiala v poljubni točki procesa
- Šaržni proces
  - proces, ki vodi k izdelavi končnih količin materiala z obdelavo določenih količin vhodnih materialov z urejenim nizom procesnih aktivnosti preko končnega časovnega intervala, pri čemer se uporablja en ali več kosov opreme
  - šaržni procesi so nezvezni procesi; niso niti diskretni niti zvezni, pač pa vključujejo značilnosti obeh tipov proizvodnih procesov

## Opis procesa po standardu S88



- Standard uvaja referenčne modele za opis šaržne proizvodnje in vodenja šaržnih procesov
- Osnova so trije hierarhično zgrajeni modeli
  - procesni model (Process model)
  - fizični model (Physical model)
  - model postopkovnega vodenja (Procedural control model)
- Modele lahko
  - krčimo oz. poenostavljamo
    - elemente modelov, ki v dani aplikaciji niso potrebni, lahko izpustimo, z izjemo nekaj ključnih gradnikov
  - razširjamo
    - lahko dodamo nove elemente, če ne kršimo integritete osnovnih odnosov med elementi modela

## Procesni model

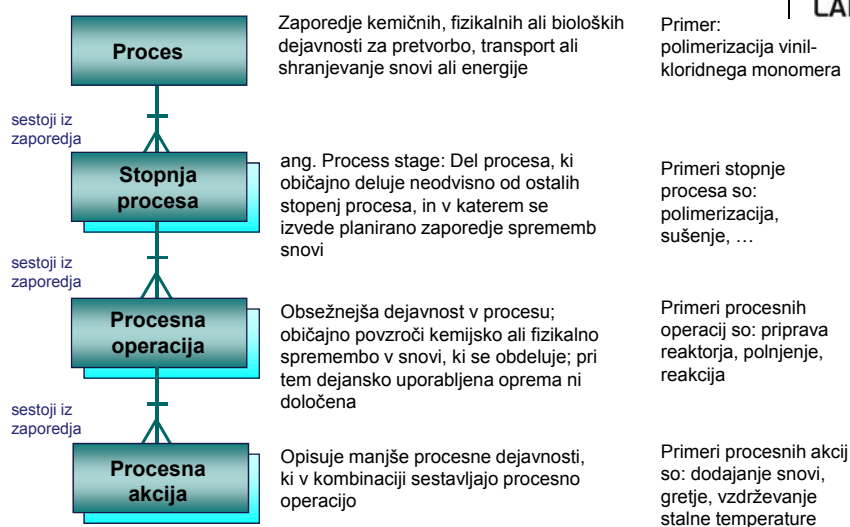
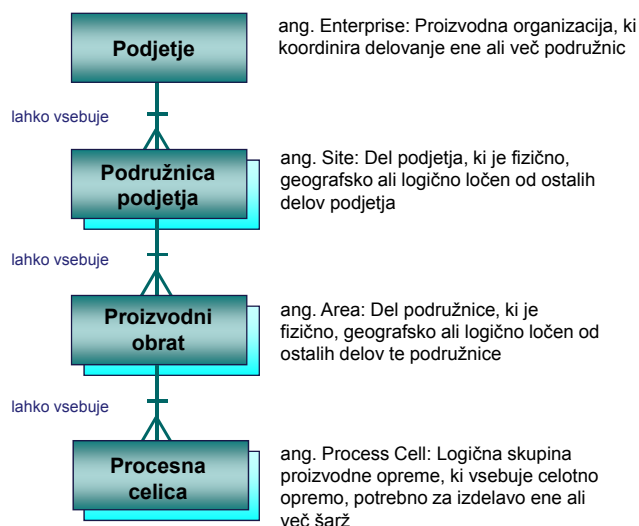
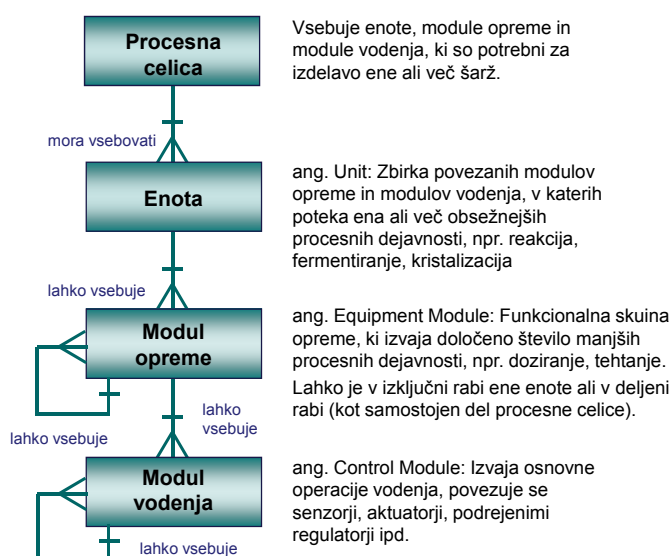


Diagram entiteta-razmerje (ER)

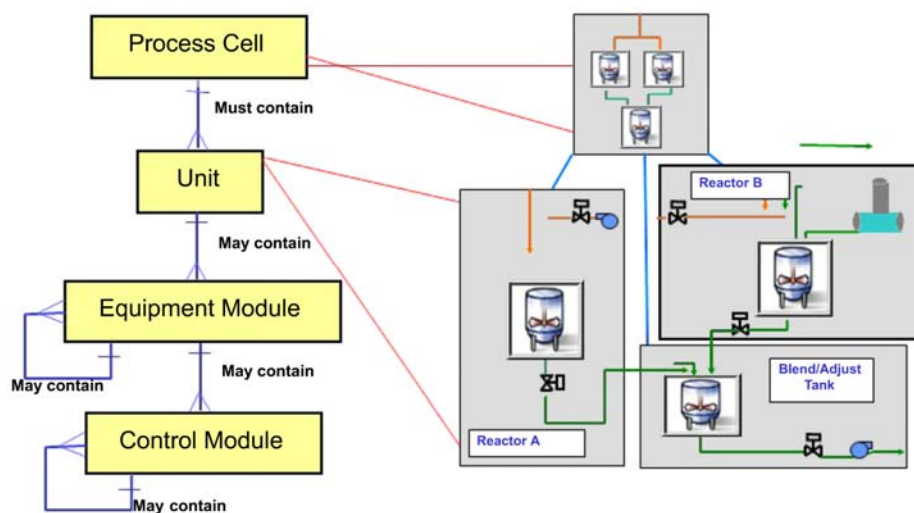
## Fizični model (zgornji nivoji)



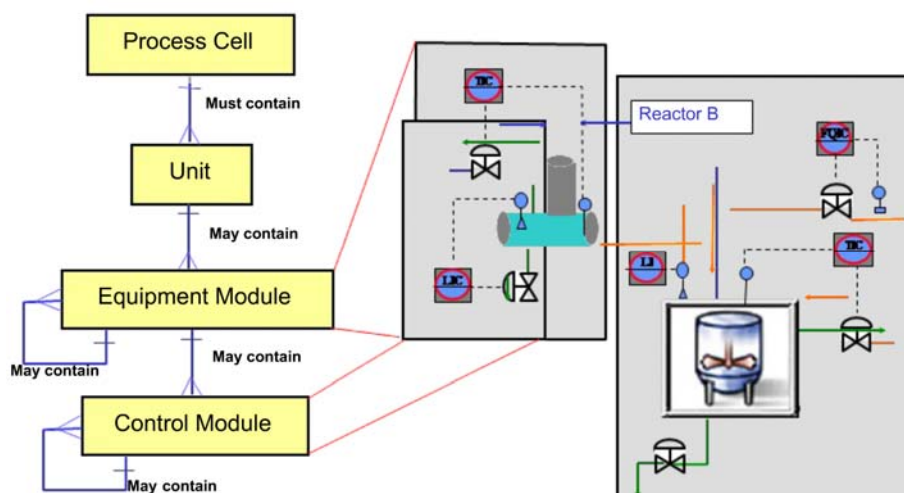
## Fizični model (spodnji nivoji)



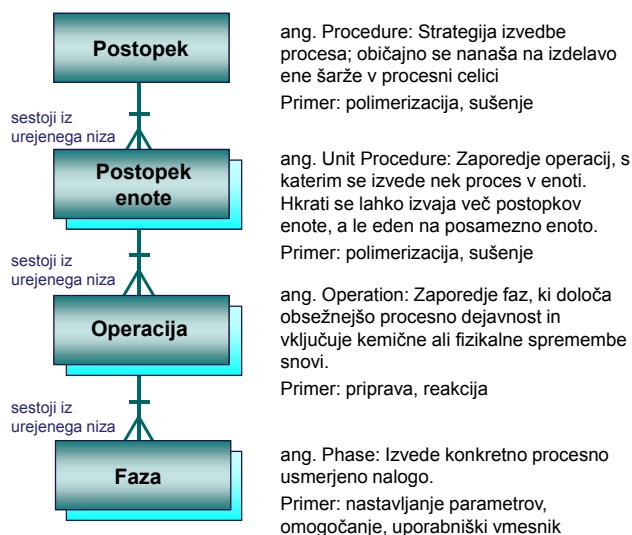
## Fizični model



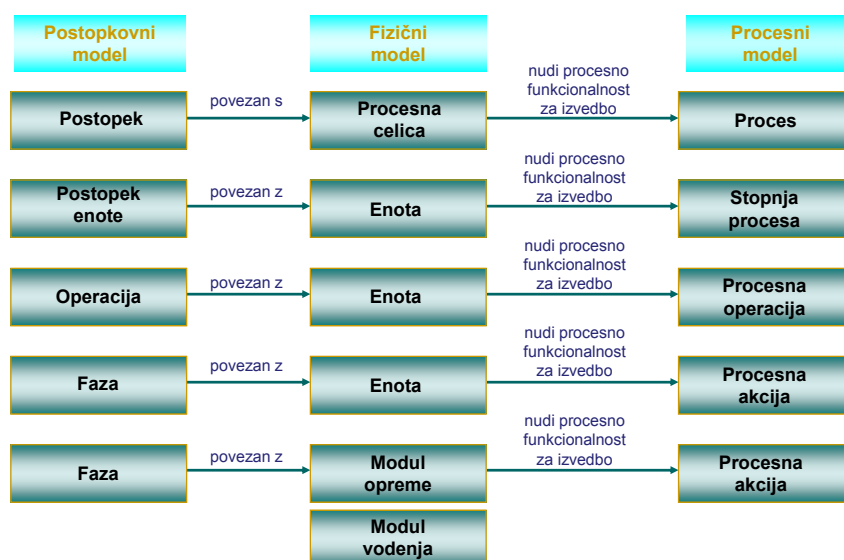
## Fizični model



## Model postopkovnega vodenja

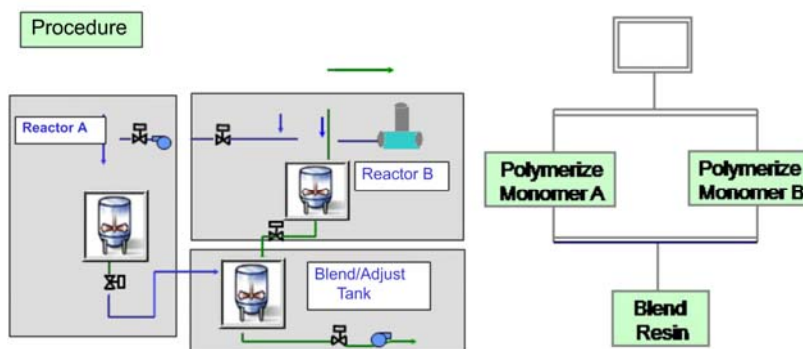


## Zveze med modeli

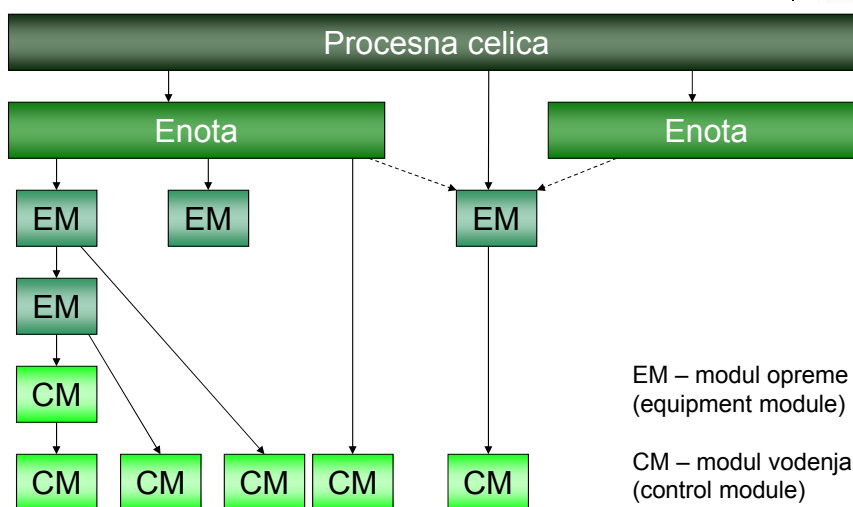




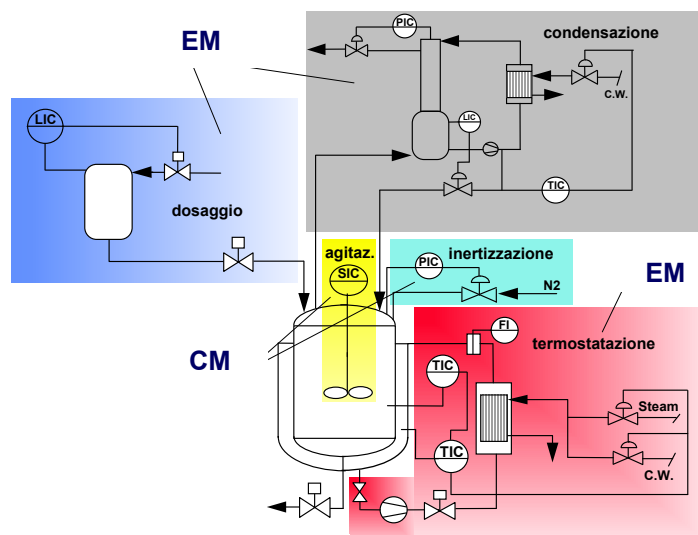
## Primer postopka



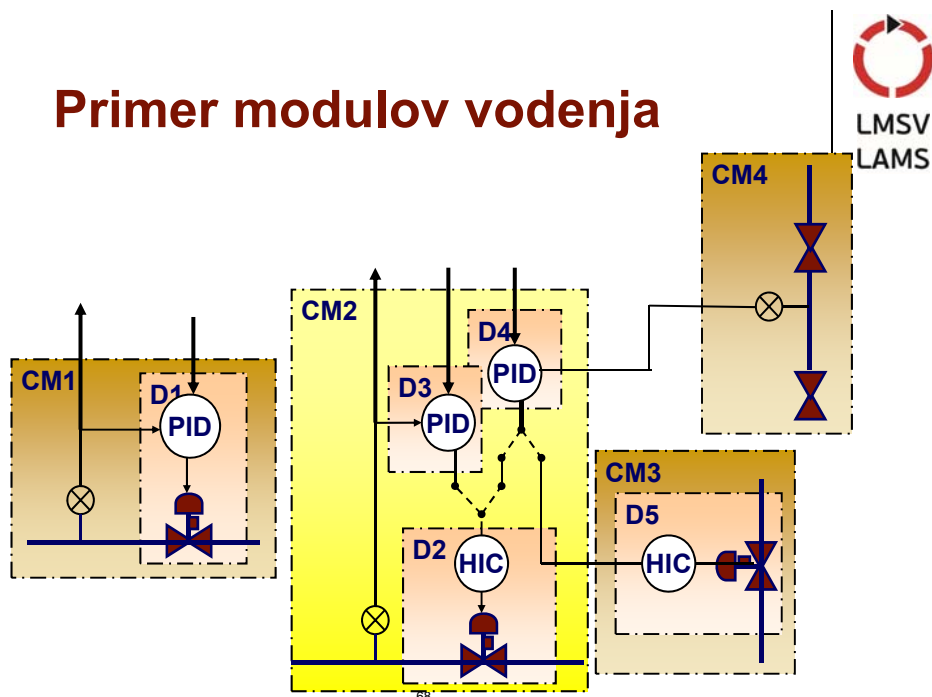
## Primer fizičnega modeliranja



## Primer strukturiranja enot



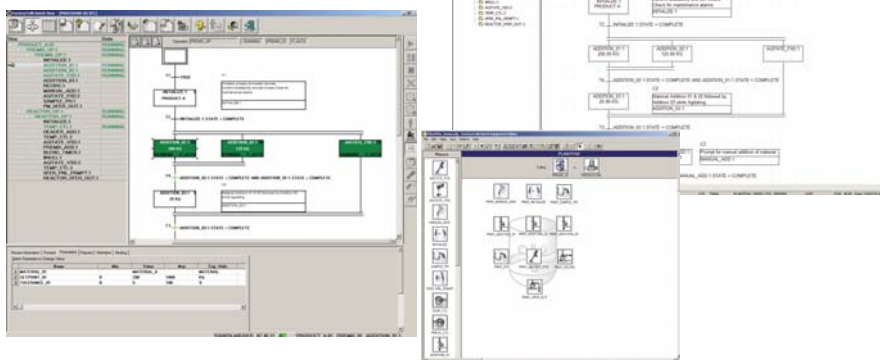
## Primer modulov vodenja



## Programska orodja za podporo standardu S88



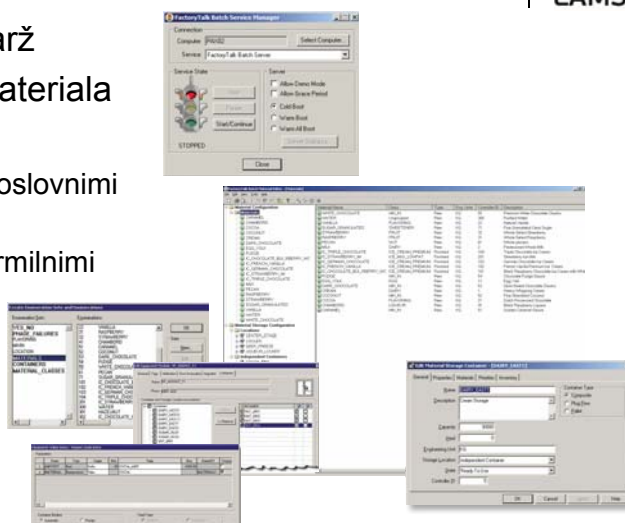
- Urejanje receptur
- Konfiguracija opreme
- Spremljanje izvajanja



## Programska orodja za podporo standardu S88



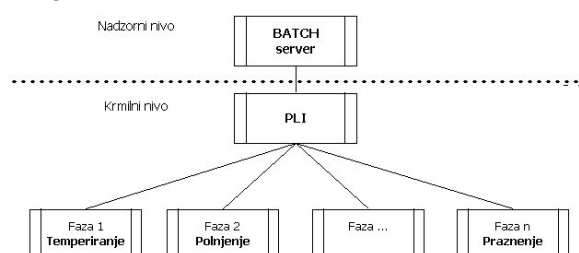
- Upravljanje šarž
- Upravljanje materiala
- Drugo
  - povezava s poslovnimi sistemi
  - povezava s krmilnimi sistemi
  - simulacija
  - arhiviranje



## Izvedba vodenja po standardu S88



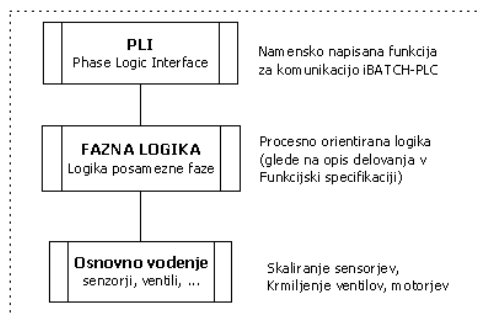
- Nadzorni nivo – šaržni strežnik
  - program na osebnem računalniku (PC)
  - skrbi za poganjanje faz v skladu z recepturno logiko
- Krmilni nivo
  - vključuje vmesnik PLI (angl. Phase Logic Interface) in fazno logiko



## Izvedba krmilnega nivoja



- Program na krmilnem nivoju je hierarhično organiziran v tri skupine
  - vmesnik PLI
  - fazna logika
  - osnovno vodenje
- Vsi deli programa se izvedejo v vsakem krmilnem ciklu



## Vmesnik PLI

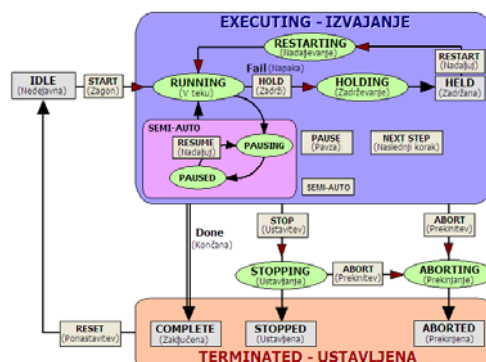


- Povezava med šaržnim strežnikom in krmilnim nivojem
  - PLI sprejema ukaze od šaržnega strežnika, jih posreduje fazam in v nasprotni smeri vrača statuse faz nazaj šaržnemu strežniku
  - narejen je namensko za proizvajalca šaržnega programskega orodja
  - funkcionalnost vmesnika PLI je določena s standardom
- PLI omogoča, da so s strani šaržnega strežnika vse faze videti enako
  - sprejemajo natančno določen nabor ukazov in vračajo vnaprej določen nabor statusov

## Vmesnik PLI



- Skrbi, da se faze pravilno izvajajo
  - v skladu s predpisanim diagramom prehajanja stanj
  - obnašanje v določenem stanju je določeno s fazno logiko



## Fazna logika



- Del programa, ki skrbi za vklope aktuatorjev, da lahko izvede procesno orientirano akcijo
- Fazna logika je
  - definirana z opisom procesa
  - vsebuje korake za izvedbo akcije, kot so: odpri ventil, vklopi črpalko, resetiraj števec ...
- Fazna logika ne vsebuje podrobnosti krmiljenja posameznih elementov
  - uporablja vnaprej pripravljene manjše programske celote – elemente osnovnega vodenja
  - z uporabo osnovnega vodenja fazno logiko razbremenimo podrobnosti o gradnikih vodenja

## Osnovno vodenje



- Najmanjši moduli krmilne logike
  - izvajajo osnovne funkcije sistema vodenja
    - krmiljenje dvopoložajnega ventila, skaliranje analognega senzorja, krmiljenje črpalke, alarmiranje v primeru termične zaščite ...
- Fazna logika tretira elemente osnovnega vodenja kot objekte s svojimi vhodi in izhodi
  - fazna logika ne postavlja direktno izhodnih signalov, temveč objektom pošilja zahteve za izvedbo akcije
  - isti element osnovnega vodenja lahko sprejema ukaze različnih faz, pomembna je pravilna izvedba

## Osnovno vodenje



- Primer povezave fazne logike in osnovnega vodenja
  - fazna logika postavi ukaz "odpri ventil" elementu "Ventil s povratno signalizacijo"
    - logika osnovnega vodenja elementa poskrbi za postavitev izhodnega signala in po določenem času preko povratnega signala preverja, ali se je ventil res odprl
    - če se ventil ni odprl, osnovno vodenje elementa javi alarm
    - fazna logika reagira na ta alarm (ne da bi se ukvarjala s povratnim signalom končnega stikala)

## PackML



- Uporaba konceptov standarda S88 pri krmiljenju diskretnih procesov
- Razvoj pod okriljem Packaging Machine Language (PackML) group
- Definira
  - tipe stanj strojev (aktivna, čakalna, dualna)
  - model prehajanja stanj, opise stanj in prehodov
  - načine delovanja strojev
  - del programa, ki upravlja z načini delovanja
  - spremenljivke za povezavo z višjimi nivoji

## Prednosti PackML za uporabnike



- Konsistentna operativna stanja in načini delovanja
  - omogočajo analizo vzrokov nastanka problema
- Uniformno upravljanje in operatorski vmesnik strojev
- Spremenljivke, ki omogočajo merjenje učinkovitosti strojev
- Hitrejša integracija strojev
- Krajši zastoji
- Manjši skupni stroški
- Manjša potreba po nadzornih sistemih
- Možna prerazporeditev strojev
- Lažja validacija delovanja

## Prednosti PackML za razvijalce strojev



- Hitrejši razvoj
- Večja ponovna uporabljivost programske kode
- Krajši čas odstranjevanja napak v programih
  - robustno programiranje
- Večja fleksibilnost
- Manj usposabljanja uporabnikov
- Lažja podpora prodaji
- Konsistentne zahteve naročnikov
- Razločevanje intelektualne lastnine
- Omogoča osredotočanje na inovativnost

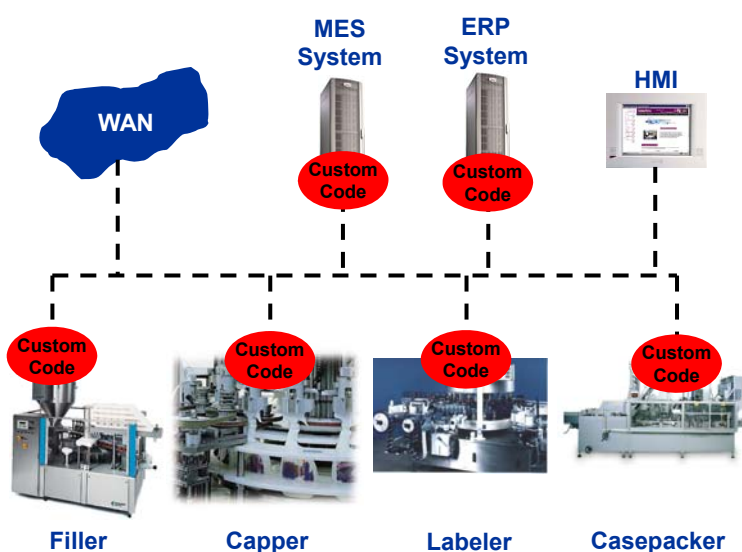


## Prednosti PackML za ponudnike opreme za vodenje

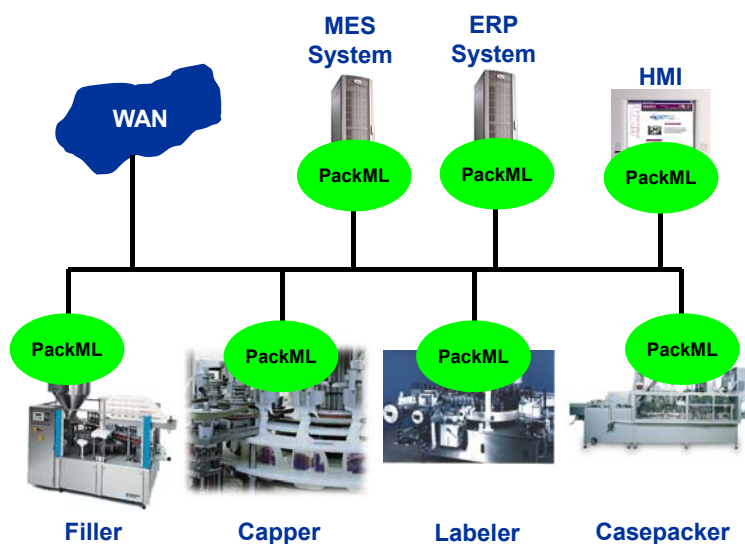


- Manj usposabljanja uporabnikov
- Uniformne knjižnice
  - lažje sprejemanje s strani razvijalcev
  - hitrejše širjenje
- Več priložnosti za plasma opreme/manjša odvisnost od uporabniških standardov
  - odločanje temelji na učinkih in zmogljivosti opreme
- Razločevanje opreme po značilnostih

## Klasičen način izvedbe



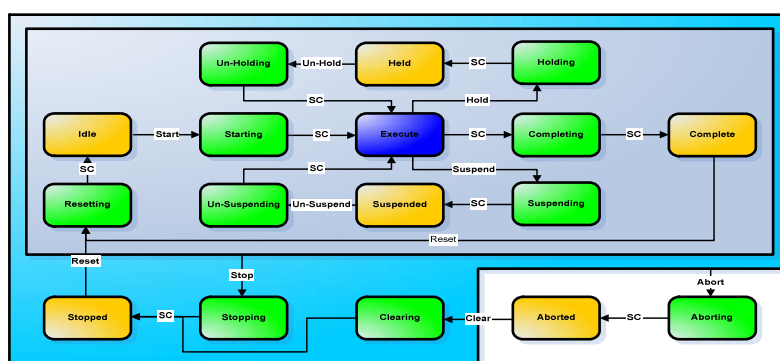
## Izvedba v skladi s PackML



## Stanja PackML



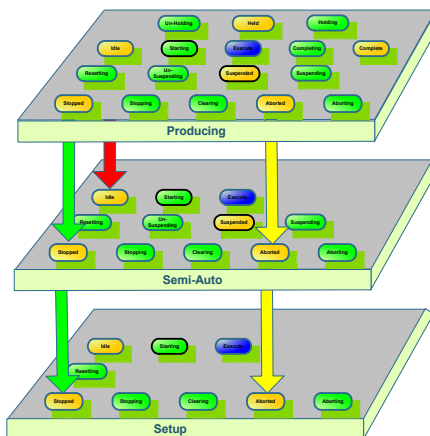
- Poenoten operatorski vmesnik (HMI)
- Konsistentne informacije
- Konsistentno prehajanje stanj



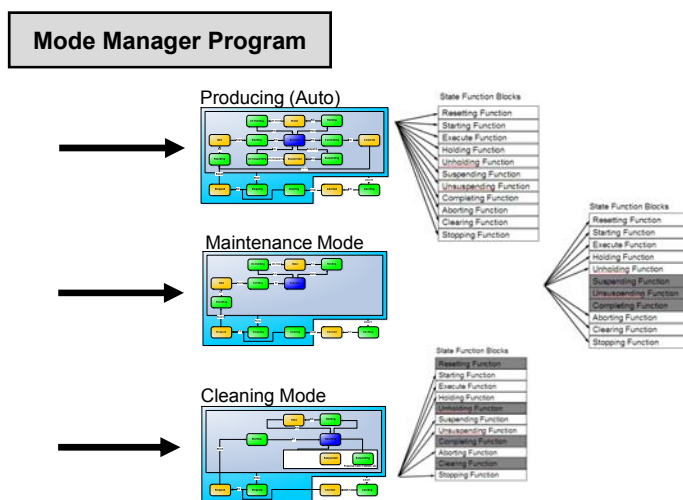


## PackML in načini delovanja

- Krmilni način enote
  - Unit Control Mode
  - funkcionalni način
- Postopkovni način enote
  - Unit Procedural Mode
  - način izvajanja
- Združljivost z S-88/ M2P (machine-to-people)



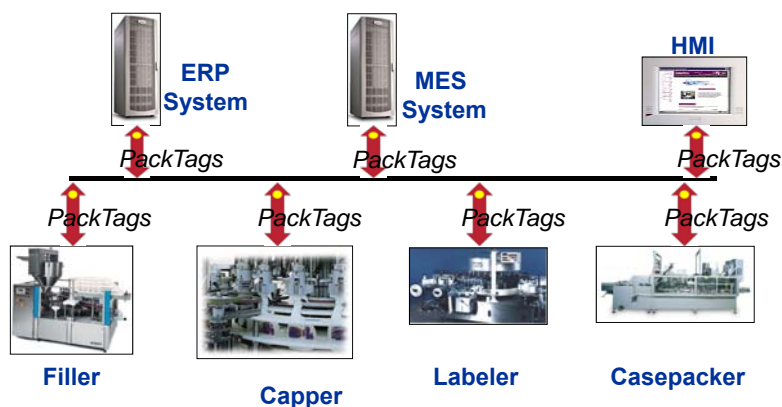
## PackML in načini delovanja



## PackML in povezava z višjimi nivoji vodenja



- Standardizirani, fleksibilni podatkovni nizi
- Poenostavljen vmesnik do poslovnih sistemov



## Primer razgradnje procesne enote v skadu s PackML



STATION 1 . DISTRIBUTION STATION						
STEP 1	STEP 2	STEP 5	STEP 3	STEP 4	STEP 6	STEP 7
Modes	States	EM's	Operation steps	Cmd	Equipment Steps	CM's
Production / Manual	Resetting	EM_DISTRIBUTOR EM_ARM	Move to initial position Move to initial position	Reset signal Reset signal	Put piston backward Put release state	CM_SIMPLE_PISTON CM_DBLE_PISTON
	Starting	EM_DISTRIBUTOR EM_ARM	Enable Cmd Enable Cmd	Start Start	Put arm on the left Enable piston Enable holding Enable rotation	CM_SIMPLE_PISTON CM_DBLE_PISTON CM_SIMPLE_PISTON
	Execute	EM_DISTRIBUTOR EM_ARM	Start piston movement Start arm movement	Starting State Complete Starting State Complete	Put piston forward Put arm on the right Suction hold the WP Put arm on the left Suction release the WP	CM_SIMPLE_PISTON CM_DBLE_PISTON CM_DBLE_PISTON CM_DBLE_PISTON CM_DBLE_PISTON
	Idle			Resetting State Complete		
Stopped				Stopping State Complete		
	Stopping	EM_DISTRIBUTOR EM_ARM	Stop Cmd Stop Cmd	Stop signal Stop signal	Stop Piston Stop Arm	CM_SIMPLE_PISTON CM_DBLE_PISTON
	Clearing	EM_DISTRIBUTOR EM_ARM	Reset faults Reset faults	Clear signal Clear signal	reset piston reset suction gripper reset rotation arm	CM_SIMPLE_PISTON CM_DBLE_PISTON CM_DBLE_PISTON
Aborted				Aborting State Complete		
Aborting	EM_DISTRIBUTOR	Abort Cmd	Abort signal	Abort signal	Abort Piston movment	CM_SIMPLE_PISTON
	EM_ARM	Abort Cmd	Abort signal	Abort signal	Abort Arm movment	CM_DBLE_PISTON