

UNIVERZA V NOVI GORICI
POSLOVNO-TEHNIŠKA FAKULTETA

DIPLOMSKO DELO

GRAFIČNI UPORABNIŠKI VMESNIKI ZA NADZOR VALJARSKE LINIJE

Jernej Bratina

Mentor: prof. dr. Juš Kocijan

Nova Gorica, 2007

ZAHVALA

Zahvaljujem se vsem, ki so mi stali ob strani pri pisanju diplomske naloge, predvsem pa mentorju prof. dr. Jušu Kocijanu, podjetju Systec ter domačim za potrpljenje in vzpodbude.

KLJUČNE BESEDE

valjarska linija

nadzorni sistem

SCADA aplikacija

SCADA programsko orodje

KEY WORDS

Rolling line

Supervisory system

SCADA application

SCADA program tool

POVZETEK

V diplomski nalogi smo podali teoretske osnove izgradnje uporabniških vmesnikov in se osredotočili na gradnjo nadzornega sistema oziroma njegovega grafičnega uporabniškega vmesnika (ang. Supervisory Control And Data Acquisition – v nadaljevanju SCADA) valjarske linije Federal Mogul. Za uspešno načrtovanje je bilo potrebno poznavanje valjarske proge. Nadzorni sistem smo izdelovali s programom Wonderware InTouch 8.0, v katerega smo morali predhodno izdelano bazo podatkov uvoziti z uporabo internega Wonderwarovega orodja. Sama SCADA aplikacija je izdelana po določenih internih standardih in zahtevah, ki jih je posredoval nemški kupec. Zgrajena je z uporabo funkcij za izgradnjo, in sicer iz zbiranja procesnih podatkov, prikazov, alarmiranja, tvorjenja alarmnih poročil. Gradnja se je končala s postavitvijo komunikacije med SCADO in krmilnikom ter na koncu s tako imenovanim preliminarnim preizkusom.

ABSTRACT

In this work we first present aspects of development graphic user interfaces. Then, we focus on developing SCADA (Supervisory Control And Data Acquisition – further SCADA) application, which control and monitor rolling line called Federal Mogul. For successful design we had to know how rolling line works. SCADA application was designed with Wonderware SCADA tool InTouch 8.0. Before developing our application, we have inserted data base with internal Wonderware tool. SCADA application was developed according to the costumer requests and with following functions, which are: colleting process data, screen designs, alarming, generating alarm reports. Development was finished with establishing communication between SCADA application and PLC. At the end we tested SCADA application with preliminary testing.

KAZALO

KLJUČNE BESEDE.....	III
KEY WORDS.....	III
1 UVOD.....	1
2 NAPRAVE ZA KOMUNIKACIJO S ČLOVEKOM	3
2.1 Interaktivne vhodne naprave	3
2.2 Interaktivne izhodne naprave	5
2.3 Interaktivne vhodno-izhodne naprave.....	8
3 GRAFIČNI UPORABNIŠKI VMESNIKI.....	11
3.1 Psihologija človeka in računalnikov	11
3.2 Grafični prikaz informacij.....	12
3.3 Človeški faktor in uporabnost.....	14
3.4 Objektno orientirani uporabniški vmesniki.....	16
3.5 Objekti in razredi.....	16
4 PREGLED SKOZI UPORABNIŠKI VMESNIK	17
4.1 Interaktivni cikel	17
4.2 Interaktivno okno	20
4.3 Proces načrtovanja vmesnika	22
5 SCADA PROGRAMSKO ORODJE	27
6 PRIMER IZDELAVE SCADA APLIKACIJE	30
6.1 Valjanje	30
6.1.1 Opis proizvodne valjarske linije Federal Mogul po valjarskih strojih.....	33
6.2 Programsko orodje SCADA Wonderware InTouch	43
6.2.1 Wonderware InTouch	43
6.3 SCADA aplikacija Federal Mogul	47
6.3.1 Glavna pravila za SCADA aplikacijo.....	47
6.3.2 Zahteve naročnika za pisanje SCADA aplikacije.....	53
6.3.3 Prikazi strani nadzornega sistema.....	55
6.3.4 Postavitev krmilnika med SCADO in PLK-jem.....	72
6.3.5 Preizkus SCADA Federal Mogul.....	73
7 ZAKLJUČEK.....	74

8	LITERATURA.....	75
---	-----------------	----

KAZALO SLIK

Slika 1: Miška in tipkovnica.....	4
Slika 2: Elektronsko pero	4
Slika 3: Ploščica na dotik	4
Slika 4: CRT monitor	5
Slika 5: LCD zaslon.....	6
Slika 6: Risalnik	7
Slika 7: Laserski tiskalnik	8
Slika 8: Zaslon na dotik izdelovalca Kontron	9
Slika 9: Interaktiven postopek načrtovanja	15
Slika 10: Ikona v logičnem stanju 1	16
Slika 11: Gumbi za navigacijo po delih procesa	17
Slika 12: Interaktivni cikel	18
Slika 13: Primerjava velikosti resolucij na zaslonu.....	21
Slika 14: Primer predstavitve akcije.....	26
Slika 15: Shematski prikaz odvijalnika	35
Slika 16: Shematski prikaz (stranski ris) odvijalnika	35
Slika 17: Stisljivi in ravnalni valji	36
Slika 18: Shematski prikaz valja v izravnalnem stroju	37
Slika 19: Shematski prikaz akumulatorja	38
Slika 20: Shematski prikaz mlina	40
Slika 21: Shematski prikaz navijalnika	42
Slika 22: Shematski prikaz (stranski ris) navijalnika	42
Slika 23: program DB Load	46
Slika 24: Polja za vpis in izpis podatkov	49
Slika 25: Status stroja	50
Slika 26: Statusi paketov	50
Slika 27: Primeri električnih simbolov.....	51
Slika 28: Odzivnost uporabnika na alarme.....	52

Slika 29: Organizacija strani.....	55
Slika 30: Videz strani.....	56
Slika 31: Prijavno okno.....	57
Slika 32: Glava SCADA aplikacije	57
Slika 33: Noga SCADA aplikacije	58
Slika 34: Navigacijske ikone	58
Slika 35: Pregled celotne linije	60
Slika 36: Pregled vhoda linije na SCADA aplikaciji.....	63
Slika 37: Pregled izhoda linije.....	65
Slika 38: Lista varnostnih vrat	66
Slika 39: Skupek strani dovoljenj	67
Slika 40: Stran skupnih alarmov	69
Slika 41: Diagnostika mreže	71
Slika 42: Nastavitve komunikacije med SCADO in PLK-jem.....	72

KAZALO TABEL

Tabela 1: Dobre barvne kombinacije.....	24
Tabela 2: Slabe barvne kombinacije.....	25
Tabela 3: Primer Excelove baze podatkov	45
Tabela 4: Strategija barvanja simbolov	48
Tabela 5: Standard barv za cevi.....	48
Tabela 6: Standard barv ventilov	48
Tabela 7: Barve alarmov	52
Tabela 8: Primer Excelove tabele v kateri so definirani alarmi	68
Tabela 9: Izvedba nalog po nivojih	70

1 UVOD

Za preučevanje grafičnih uporabniških vmesnikov smo se odločili, ker le-ti sodijo v eno najpomembnejših področij relacij med človekom in strojem, ki je dandanes pomembna prvina vsakdanjega življenja. V prihodnosti bo vedno bolj pridobivala na pomenu zaradi učinkovite komunikacije, uporabnosti, grafične dovršenosti in dostopnosti.

Z grafičnimi uporabniškimi vmesniki se srečujemo tako v gospodinjstvu kot v industriji. V gospodinjskih aparatih jih najdemo kot manj kompleksne in enostavnejše za razumevanje. V industriji pa za učinkovit nadzor nad tehnološkim proizvodnim procesom uporabniki potrebujejo zahtevnejši, dolgotrajnejši uvajalni trening razumevanja grafičnega uporabniškega vmesnika zaradi same kompleksnosti, zahtevnosti in edinstvenosti proizvodne linije.

Tukaj se lahko osredotočimo na industrijske grafične vmesnike (ang. Supervisory Control And Data Acquisition - v nadaljevanju SCADA), o katerih bo govora v nadaljevanju. Avtomatizacija proizvodnih linij postaja ena izmed prvin v gospodarstvu. Tako danes v industriji najdemo uspešna podjetja, ki pa niso uspešna samo zaradi dobre ekonomije dela ter kadra v proizvodnji, ampak je za uspeh potrebno nekaj več, kot je na primer visoko tehnološka produktivna proizvodnja.

»Naložbe v visoko produktivne visokotehnološke linije zmanjšujejo stroške proizvodnje in zato večajo kapitalske dobičke podjetij. Izboljša se kakovost izdelkov, kar posledično zmanjšuje možnost reklamacij in izgubo dobička. Avtomatizirana proizvodnja, nadzorovana z računalniki, dviguje kakovost proizvedenih izdelkov in zmanjšuje izmet.«¹

K odločitvi o temi diplomskega dela so pripomogle izkušnje iz podjetja Systec d.o.o., kjer se vsakodnevno soočamo s problemi avtomatizacije železarskih proizvodnih linij in posledično učinkovite rabe grafičnih uporabniških vmesnikov, ki morajo pri tem ustrezati določenim standardom izdelave, videza in funkcionalnosti uporabe.

¹ Mojca Kralj, Politike tehnološkega razvoja avtomobilske industrije, magistrsko delo, Univerza v Ljubljani, Fakulteta za družbene vede, str. 21

Namen diplomske naloge je nazoren prikaz izdelave SCADA aplikacije proizvodnega tehnološkega sistema. Kot cilj bi lahko opredelili poznavanje osnovnih pravil oz. napotkov za kvalitetno izgradnjo grafičnih uporabniških vmesnikov, ki so ponazorjeni na praktičnem primeru SCADA aplikacije Federal Mogul.

Diplomsko delo je zastavljeno tako, da vključuje zasnove razvoja in uporabe grafičnih uporabniških vmesnikov od osnovnih gradnikov pa do reševanja brezna vrednotenja in izvršitve ter učinkovite rabe interaktivnega okna. Najprej smo nalogo osvetlili s teoretičnega vidika, to znanje smo kasneje uporabili za primer valjarske linije Federal Mogul, ki deluje na inštitutu v nemškem Achnu.

Diplomsko delo je razdeljeno na štiri vsebinske sklope. V prvem vsebinskem sklopu prikažemo naprave za komunikacijo s človekom. Tukaj se dotaknemo interaktivnih vhodno – izhodnih naprav, pazljivost posvečamo zaslonom na dotik (ang. Touch panels). V drugem vsebinskem sklopu se osredotočimo na grafične uporabniške vmesnike, znotraj poglavja opišemo interaktivni cikel ter interaktivno okno ter začnemo s procesom načrtovanja grafičnega uporabniškega vmesnika. Tretji vsebinski sklop diplomskega dela se nanaša na SCADA programsko orodje. V tem poglavju opisujemo splošne značilnosti SCADA programskih orodij ter njihove najvidnejše proizvajalce. V zadnje in najvažnejše poglavje smo vključili primer izdelave SCADA aplikacije. Ta vsebinski sklop je razdeljen na več podsklopov. Prvi pod sklop se nanaša na valjanje pločevine ter na stroje v valjarski progi, katere krmilimo oz. jih nadzorujemo s pomočjo SCADA aplikacije. Drugi podsklop vključuje opis programskega orodja InTouch 8.0, ki smo uporabljali za izgradnjo SCADA aplikacije. Zadnji podsklop pa opisuje SCADA aplikacijo Federal Mogul z vsemi glavnimi internimi pravili, zahtevami naročnika, prikazom glavnih okenskih strani in opisom postavitve komunikacijskega krmilnika s PLK-jem (slo. Programljivi Logični Krmilnik). Zadnji podsklop smo zaključili s preliminarnim preizkusom delovanja SCADA aplikacije.

2 NAPRAVE ZA KOMUNIKACIJO S ČLOVEKOM

Pomemben del sistemov za vodenje procesov predstavljajo naprave za komunikacijo s človekom. V osnovi morajo zagotavljati tehnično podporo za dvosmeren pretok informacij, in sicer:

- pretok informacij iz procesa k uporabniku (spremljanje informacij o delovanju procesa) in
- pretok informacij od uporabnika k procesu (vplivanje operaterja na proces).

Naprave za komunikacijo s človekom delimo na dva dela, in sicer na:

- klasične naprave in
- računalniške naprave.

Med klasične naprave za komunikacijo med človekom in strojem štejemo sinoptične plošče, komandne pulte ter zvočne in svetlobne alarme.²

2.1 Interaktivne vhodne naprave

Naslednja uporabna točka, pri kateri se je potrebno ustaviti, je izbor zunanje interaktivne vhodne naprave. Na trgu je veliko naprav za tvorjenje vhoda. Vsaka izmed njih ima točno določeno obnašanje v smislu uporabniške fizične interakcije s sistemom. Med računalniške naprave za komunikacijo spadajo računalniški terminali, video zaslone, tipkovnice, tiskalniki, risalniki, zaslone, ki prepoznajo dotik prsta, miške in krmilne palice ipd.

Skoraj vsi današnji uporabniški vmesniki, temeljijo na uporabi miške (slika 1). Miška je odlična kazalna in izborna naprava, ker direktno prenaša premike uporabnikovih rok na zaslon.

² S. Srmčnik, Celostni pristop k računalniškemu vodenju procesov



Slika 1: Miška in tipkovnica

Toda miška je preokorna za risanje, skiciranje ali podpisovanje lastnega imena. Z rokami napravo zelo težko natančno premikamo v smereh, katere bi potrebovali npr. za podpis. Za take aktivnosti je boljša izbira elektronsko pero (slika 2), ki izkorišča prstne mišice. Toda pri preprostih opravilih (urejevanje besedila), pri katerih si pomagamo s tipkovnico, je držanje miške v desni ali levi roki enostavnejše, saj med delom ni potrebno uporabnikom spuščati pogleda in iskati npr. peresa - interakcija je tudi hitrejša.



Slika 2: Elektronsko pero

Poleg miške, tipkovnice in peresa je pogost interakcijski pripomoček tudi ploščica na dotik (slika 3). Le ta ima svoje prednosti, ker zavzame malo prostora (zato jo najdemo tudi na prenosnih računalnikih) ter omogoča hitro interakcijo s sistemom brez spuščanja oči z zaslona. Ima tudi slabosti, ki se poznajo v njeni majhnosti, nenatančnosti itd.



Slika 3: Ploščica na dotik

2.2 Interaktivne izhodne naprave

Slikovni spomin je dominantna arhitektura zaslonske naprave. Grafični paket, ki je del interaktivnih orodij deluje v centralni procesni enoti, ta pa določa vrednosti oz. zapolnjuje prazna spominska mesta v slikovnem spominu. Ta spomin je odložišče za podatke, ki kasneje tvorijo na grafičnem prikazovalniku sliko. Centralno procesna enota je v večini primerov sposobna brati in spreminjati podatke, ki se nahajajo v slikovnem spominu. Različna strojna oprema uporablja različne tehnike prevajanja vsebine oz. podatkov iz slikovnega spomina v vidno sliko.³

Zaslonski na katodno cev

Zaslonski na katodno cev (slika 4) izhajajo oz. bazirajo na tehnologiji, ki izhaja iz standardnih televizijskih sprejemnikov. Ta tip naprave mora zaslonski upravljavec preslikovati slikovni spomin od zgoraj navzdol ter z leve proti desni, da prejme vrednosti slikovne točke, istočasno pa elektronski žarek preslikuje glavo zaslona po podobnem vzorcu. Zaslonski upravljavec spreminja intenzivnost elektronskega žarka glede na vrednost slikovne točke, ki jo najde v zaslonskem spominu. Tako nastane na fosforjevem zaslonu slika. V večini barvnih zaslonov najdemo tri tipe elektronskih žarkov: za vsako osnovno barvo po eden in sicer: rdečo, zeleno in modro.



Slika 4: CRT monitor

³ Dan R. Olsen, Jr, Developing User Interfaces

Fosforna slika razpade na osnovi prisotnosti fosforja. Če je prisotnost fosforja dolgotrajna, slika ne razpade v osveženem času, kar povzroča sledi hitro premikajočim se objektom. Drugi skrajni primer je prekratka prisotnost fosforja, saj slika na zaslonu razpade, preden jo zaslonski upravljaivec utegne osvežiti. Uporabniki to zaznamo kot tresenje zaslonske slike. Če se zaslon osvežuje s hitrostjo 30 slik na sekundo oz. 30Hz, uporabniki ne zaznamo tresenja ali sledi. Danes monitorji omogočajo osveževanje zaslonov v frekvenci 100Hz, resolucija pa se giblje od 640×480 slikovnih točk naprej.

Zaslone na tekoče kristale (LCD)

Zaslone na tekoče kristale (LCD, slika 5) danes najdemo tako pri osebnih računalnikih kot tudi pri prenosnih računalnikih. Uveljavili so se zaradi ploskosti in majhne porabe energije. Pod vplivom električne napetosti se spreminjajo polarizacijske lastnosti kristalov. S polarizacijo in s pomočjo polarizatorskih filtrov, ki skrbijo, da kristali postanejo prozorni oz. neprozorni, se na zaslonu prikaže slika.

Cenejši LCD-ji za svoje delovanje uporabljajo tehnologijo pasivne matrike, kjer se slikovna točka in obseg kontrole kristalov nahajata na periferiji zaslona. Na tak način dobimo sliko brez tresenja, toda čas, ki je potreben za spremembo barve slikovne točke, je bistveno krajši, kot je sprejemljivih 30 slik na sekundo. Primerni so za stoječo sliko, z malo ali skoraj nič premikajočih se objektov.

Dražji LCD-ji uporabljajo za delovanje aktivno matriko, kjer kontrola obsega kristalov direktno vpliva na slikovno točko. S tem je omogočen boljši kontrast slike ter hitrejši čas izmenjevanja slikovnih točk.



Slika 5: LCD zaslon

Ravno tako kot zasloni s katodno cevjo tudi tekoče kristalni zasloni uporabljajo zasloni spomin, ki temelji na podobni tehnologiji. Edina razlika je v zaslonem krmilniku, saj pri tekoče kristalnih zaslonih le ta skrbi za neprepustnost kristalov. Podobna tehnologija zaslonkega krmilnika se uporablja tudi pri plazemskih zaslonih.

Papirni prikazovalniki

Tiskalniki (slika 6) in risalniki (slika 7) so naprave, ki so indirektno vpletene v uporabniški vmesnik. Čeprav lahko rečemo, da se tehnike programiranja in programski dizajn uporablja tako za zaslonke kot za papirne prikazovalnike. Glavna razlika med papirnimi in zaslonkimi prikazovalniki je v tem, da centralna procesna enota pri papirnih prikazovalnikih nima direktnega dostopa do slikovnega spomina, zato je komunikacijska hitrost med CPU-jem in slikovnim spominom zelo pomembna.

Ena izmed najstarejših naprav za papirno prikazovanje je risalnik. Naprava je bazirana na poteznem slikovnem modelu. Koordinate poteznih objektov, ki so poslane v risalnik so fizične. Risalnik nato premika pero po papirju in riše črte. Ta proces ima slikovne omejitve, saj s črtami zelo težko predstavimo realistične slike.



Slika 6: Risalnik

Moderni laserski tiskalniki so ravno tako kot zasloni bazirani na slikovnem spominu. Poglavitna težava nastane, ker pri 600 pikah na inč (dpi), slikovni spomin potrebuje približno 3.6MB spominskega prostora za vsako stran (A4 format list). Prenajanje tolikšne količine informacij je preveliko, zaradi tega laserski tiskalniki v splošnem uporabljajo t.i. regijski model. Najbolj popularen je »PostScript«, ki je baziran na FORTH programskem jeziku. Ta programski jezik omogoča tiskalniku, da sam zapolni

slikovni spomin s točno določenimi, razdeljenimi »regijskimi« podatki. Ta metoda omogoča tiskanje zahtevnih slik, ne da bi bila komunikacija prezasedena, saj podatki prihajajo v tiskalnik paketno oz. regijsko.



Slika 7: Laserski tiskalnik

V primeru fotografije je »regijski« model neuporaben. Velika večina laserskih tiskalnikov lahko sprejme slike, ki so bazirane na slikovnih točkah. Le te so prenesene in prevedene v tiskalniški slikovni spomin. Take slike oz. fotografije so navadno nižje resolucije, kot je maksimalna resolucija tiskalnika, kar posledično zmanjšuje oz. odpravlja napake v komunikacijskem kanalu.

Podatke iz slikovnega spomina tiskalnik na papir prenaša preko laserja, ki na papir »piše« z elektrostatičnim nabojem na t.i. bobnu ali pasu. Toner (črnilo v prahu) privlači elektrostatični naboj, ki omogoča prijemanje črnila na podlago, kjer je prisoten električni naboj. V praksi pomeni, da papir, na katerega se »lepi« črnilo v prahu, potuje skozi pas oz. boben laserskega tiskalnika. Tehnologijo tiskanja dopolnjujejo brizgalni tiskalniki, ki ravno tako uporabljajo za svoje delovanje slikovni spomin, sposobni pa so tiskanja v barvah, na osnovi barvne sublimacije ipd.

2.3 Interaktivne vhodno-izhodne naprave

Zaslone na dotik

Pri opisu računalniških naprav za komunikacijo se bomo posebej osredotočili na zaslone (slika 8), ki prepoznajo dotik prsta.



Slika 8: Zaslon na dotik izdelovalca Kontron

Zaslone občutljivi na dotik predstavljajo alternativo ostalim napravam, ki komunicirajo s človekom. V tem primeru poseben elektronski sistem prepozna položaj dotika prsta na zaslonu in omogoča izvršitev določenega ukaza, ki je prikazan na navidezni tipki pod točko dotika prsta. Namen zaslonov na dotik je povečanje hitrosti vnosa podatkov v računalnik.

Danes so zasloni na dotik do uporabnika bolj prijazen in robusten nadomestek računalniške tipkovnice in miške. Z zasloni na dotik je dovoljena neizkušenim uporabnikom enostavna interakcija s kompleksnimi programi, ne da bi se le ti tega zavedali. S takimi zasloni je prišel računalnik tudi na mesta, kjer prej njegova postavitve ni bila možna zaradi omejitev, ki jih imata tipkovnica in miška.

Osredotočimo se lahko na pet komponent, ki sestavljajo zaslon na dotik:

- Senzor za dotik, ki je lahko: kapacitiven, uporoven, infrardeč, (ang. Surface Acoustic Wave SAW, slo. površinski akustični valovi, ang. Grounded Acoustic Wave GAW, slo. zaprti akustični valovi) itd.
- Zaslon: zasloni na katodno cev, tekoče kristalni zasloni ali plazma, na katerega je mogoče pritrditi senzor za dotik.
- Krmilnik: omogoča, da senzor deluje kot periferna naprava.
- Programski krmilnik: s programskim krmilnikom je omogočena komunikacija med krmilnikom in računalniškim operacijskim sistemom.

- Osebni računalnik (ang. Personal Computer – PC), ki komunicira in vodi z zaslonom na dotik.

Kapacitivni zasloni na dotik

Analogni kapacitivni zasloni na dotik so narejeni iz prevodne folije, ki prekriva prozorni stekleni senzor. Električna napetost se pojavi na vseh štirih vogalih zaslona in jo najdemo na obeh X-Y koordinatah. Ko zaslon ni v uporabi, elektrode razporedijo električno napetost po zaslonu v urejenem vzorcu. Z dotikom prsta na zaslon polje prepozna motnjo. X-Y koordinate dotika prsta, se prenesejo preko krmilnika in programskega krmilnika na PC serijski vhod.

Uporovni zasloni na dotik

Uporovni zasloni na dotik so narejeni iz prekrivajočega zgornjega sloja debeline 1/10000 milimetra in prevodnega poliestra, ki leži zelo blizu prevodnega steklenega sensorja. Ko se uporabnik dotakne trdega, proznega, prevodnega poliestra na zaslonu, se vzpostavi kontakt s stabilnim steklenim senzorjem. Po vzpostavitvi kontakta se električna napetost pojavi na vseh štirih robovih zaslona v sorazmerju z razdaljo od točke dotika do roba. Kot pri kapacitivni tehnologiji se tudi pri uporovnih zaslonih koordinate dotika prenesejo preko krmilnika in krmilnika na PC serijski vhod.

Za razliko od kapacitivnih lahko pri uporovnih zaslonih na dotik tvorimo vhod tudi z rokavicami, nohtom, peresom ali celo s kreditno kartico. Ker se vhod dotika na zaslonu tvori s pomočjo prevodnega poliestra, zaslon na dotik ni občutljiv na praske oz. na pretrgan zgornji sloj. Mnogostranost take tehnologije izdelave zaslona na dotik omogoča delovanje zaslona v težkih delovnih okoljih (npr. prašno delovno okolje).

Infrardeči zasloni na dotik

Infrardeči zasloni na dotik so narejeni kot okvir, ki ga je možno postaviti pred zaslon na katodno cev. Sestavljen je iz infrardečih diod, ki so razporejene po horizontali in vertikali zaslona. Pritisk na tak tip zaslona prekine infrardeči žarek, sensorji pa prenesejo koordinate prekinjenih žarkov v računalnik preko serijskega vhoda. Infrardeči

zasloni se danes redko uporabljajo, saj imajo pomanjkljivosti, kot so npr. majhna natančnost, prezgodnja zaznava dotika, problemi s paralakso.

SAW in GAW zasloni na dotik

Ti tipi zaslonov na dotik imajo podobno tehnologijo in so sestavljeni iz prekrivne prozorne folije ter sprejemnimi in oddajnimi piezoelektričnimi spreminjevalci, ki se nahajajo na vertikalni in horizontalni osi. Krmilnik na zaslonu na dotik pošilja 5MHz električni signal na sprejemni spreminjevalec, ki spreminja električne signale v mehanske nadzvočne valove znotraj stekla. Te valove usmerja po zaslonu polje reflektorjev. Reflektorji na robovih zaslona odbijajo mehanske valove v sprejemni spreminjevalec, ki spreminja mehanske valove v električne, ter tvori digitalni zemljevid površja zaslona. Ko se prst uporabnika dotakne zaslona, se majhen del valovanja absorbira. Sprejeta motnja valovanja se primerja s shranjenim zemljevidom, sprememba se nato preračuna v X, Y koordinate. Z merjenjem količine absorbiranega signala lahko definiramo tudi Z koordinato.

3 GRAFIČNI UPORABNIŠKI VMESNIKI

3.1 Psihologija človeka in računalnikov

Po pregledu strojev v valjarski liniji se nam zdi potrebno usmeriti nekaj pozornosti tudi na psihologijo človeka in računalnikov, saj grafični uporabniški vmesniki temeljijo na komunikaciji med človekom in strojem oz. računalnikom, kjer ugotovimo, da računalnik še ne »misli« kot človek, kljub temu pa se mora človeškemu »mišljenju« približati, da omogoči učinkovito rabo operacij. Če hočemo, da je računalnik učinkovit pripomoček za človeka, moramo poznati človekove zaznave ter njegovo procesiranje informacij. Zaznavanje je umeščanje novih izkušenj v korelacijo z izkušnjami in pričakovanji

Procesiranje informacij se deli na tipe človeškega spomina. Le ti so:

- senzorni spomin,
- kratkotrajni spomin,

- dolgotrajni spomin.

Senzorni spomin sprejema veliko količino informacij, ki se jih človek ne zaveda. Na zaznavanje vplivajo spremembe, ki se nanašajo na zvok, svetlobo, barve ter gibanje (animacija na monitorju, film itd.).

Kratkotrajni spomin hrani 7 ± 2 podatka (besede, pojme, števila) za čas približno 30 sekund.

Dolgotrajni spomin ima informacije kodirane. Ta spomin lahko hrani do 100.000 podatkov.

Vmesnik pomaga človeku pri razpoznavanju podatkov pri delu z računalnikom.

3.2 Grafični prikaz informacij

Grafični prikaz informacij je zelo dobrodošel način manipulacije s podatki, saj človek sprejema približno 80 % informacij vizualno. Zaradi tega ima grafično prikazovanje informacij sledeče prednosti:

- hitrejše razpoznavanje,
- hitrejše učenje,
- hitrejša uporaba orodij,
- minimalno abstraktno mišljenje,
- manj napak,
- takojšnja povratna informacija,
- predvidljiv odziv,
- ni jezikovnih ovir,
- atraktivnost,

- manj stresa.⁴

Kaj je grafični uporabniški vmesnik

» Grafični uporabniški vmesniki GUI – (ang. Graphic User Interface) so največkrat povezani z operacijskim sistemom, ni pa nujno. Nastali so kot nadgradnja operacijskih sistemov, dandanes pa so večinoma vanje kar neločljivo vgrajeni. Svoj grafični vmesnik ima lahko tudi vsak posamezen program, in sprva so ga nekateri tudi imeli, a za uporabnike je seveda zelo koristno, če vsi programi uporabljajo enakega, pa tudi programerjem to močno olajša delo.«⁵

Obstoječi grafični sistemi

Danes, ko pričakujemo oz. testiramo zadnje različice beta verzije operacijskega sistema Windowsov Vista, si sploh ne moremo predstavljati, kako so grafični uporabniški vmesniki izgledali pred 50 leti. Ne predstavljamo si več računalniških strokovnjakov, kako hodijo okoli velikih analognih zapletenih strojev, takrat imenovanih računalnikov. Obstajali pa so vizionarji, ki so vedeli oz. predvidevali, v katero smer se bo tehnologija računalnikov razvijala. Pričakovali so, da se bodo morali ti veliki stroji čez nekaj let zmanjšati ter postati enostavnejši za uporabo, za kar so v določeni meri poskrbeli tudi uporabniški vmesniki oz. njihovi razvijalci. Tako dandanes poznamo kar nekaj različnih tipov grafičnih uporabniških vmesnikov, ki so pravi razmah uporabe doživeli v sredini osemdesetih let. Na tej točki lahko naštejemo nekaj grafičnih uporabniških sistemov, kjer nekateri, predvsem novejši temeljijo na operacijskih sistemih z enakimi imeni ter njihove proizvajalce, ki so jih razvijali od začetka osemdesetih let pa do danes. Ti vmesniki so sledeči:

- Macintosh (1984, razvijalec Apple, prvi široko sprejet grafični sistem)
- Microsoft Windows (1985, razvijalec Microsoft, alternativa ukaznemu načinu operacijskega sistema MS-DOS)

⁴ Franc Jager, Uporabniški vmesniki

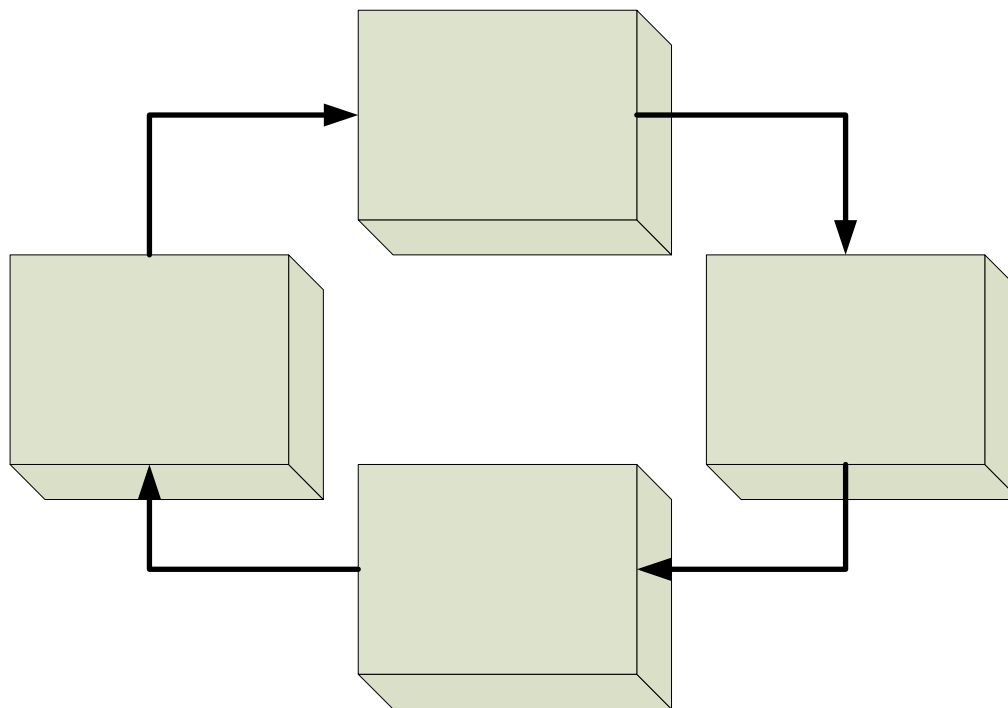
⁵ Nikolaj Pečenko, Okna, ikone, menuji & miške, Monitor, januar 2006, str 114

- DECwindows (1987, razvijalec DEC temelji na UNIXU, koncept strežnik odjemalec)
- OS/2 Presentation Manager (1987 Microsoft in IBM, alternativa ukaznemu načinu operacijskega sistema MS-DOS)
- NeXTStep (1988, prva simulacija 3D zaslona)
- OPEN LOOK (1989, razvijalec AT&T in Sun Microsystems, temelj je UNIX, inovativne lastnosti)
- OSF / Motif (1989, HP in DEC, Simulacija 3D zaslona)
- Microsoft Windows (1989, razvijalec Microsoft, Windows 3.0)
- OS/2 Workplace Shell (1992, razvijalec IBM, grafični uporabniški vmesnik za OS/2)
- Microsoft Windows (1992, razvijalec Microsoft, Windows 3.1)
- Windows 95, 98, OS/2 Warp (1995-2000)
- Windows 2000, XP, ukazni način X Windows, ki predstavlja AIX UNIX, LINUX (od 2000 naprej)

3.3 Človeški faktor in uporabnost

Primarni cilj uporabniškega vmesnika je pospešitev služenja računalnika človeku. Kot pomemben koncept bi lahko izpostavili, da so programerji samo programerji. Svoj poklic so izbrali zaradi potrebe po novih programerskih izzivih, medtem ko ostali uporabniki računalnika nimajo takih interesov. Največji problem za programerja je razumevanje potreb uporabnika vmesnika. Kar morajo programerji vedeti, je, da obstajajo ljudje s poklici, ki se ukvarjajo z razumevanjem potreb. Sodelovanje s takim profilom ljudi je nujno za uspešno razvijanje uporabniškega vmesnika.

Gradnja kvalitetnega uporabniškega vmesnika ni znanost. Obstaja nekaj principov in navodil, ampak ti so ali preveč splošni ali prenatanni. Za razliko od tekstovnih ali numeričnih metod pri programiranju, tukaj ni neke splošne procedure, ki bi zagotavljala kvaliteten rezultat. Prevladujoča inženirska paradigma oz. miselni vzorec za uporabniške vmesnike je ponavljajoče se načrtovanje, kot je prikazano na sliki 9.



Slika 9: Interaktiven postopek načrtovanja

Najdi upo
potrebe in

Programerji bodo ob pogledu na sliko 9 zasledili, da je ponavljajoče načrtovanje neskončno kroženje, kar v resnici tudi je. V praksi pa ponavljajoče načrtovanje prekinajo faktorji, kot so npr. ekonomika, potrpljenje vodilnih ter pretek časa, ki je določen za predajo uporabniškega vmesnika v uporabo. Toda kroženje v smislu ažuriranja se mora nadaljevati, dokler program obstaja na trgu.

Vrednotenje z uporabnikom

3.4 Objektno orientirani uporabniški vmesniki⁶

Za kar najbolj ilustrativno predstavitev objektno – orientiranih programov lahko opišemo primer ikone gumba na sliki 10, Predpostavljamo, da hoče uporabnik aktiviranje ikone z imenom »leveller«. Sama ikona mora biti programirana tako, da sprejema vhode iz miške ter hkrati vklopi kontrolno lučko, tako da uporabnik ve ali je ikona na logičnem stanju ena ali nič oz. ali je vklopljena ali izklopljena. Poleg tega mora tudi posredovati ukaz programljivemu logičnemu krmilniku PLK-ju (ang. programmable logic controller -PLC), da le ta izvede zahtevano akcijo. Z modeliranjem vsake interaktivne točke, kot objekta, ki si lahko zapomni statuse ter prejema in oddaja sporočila, dosežemo vrsto oblike obnašanja, ki je primerna za uporabniški vmesnik. Poleg izbranega gumba imamo lahko v grafičnem uporabniškem vmesniku tudi druge enake gumbе za podobne akcije. S tem pridobivamo na konsistentnosti oz. podobnost med gumbi, kar je za uporabnika dobrodošlo.



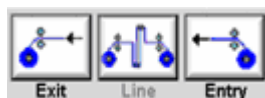
Slika 10: Ikona v logičnem stanju 1

3.5 Objekti in razredi

Objekt se obnaša kot samostojna enota znotraj računalniškega spomina. Vsak objekt ima razred, ki definira informacijo v objektu ter njegovo obnašanje. Objekti imajo polja, ki vsebujejo informacije. Polja imajo tudi določen tip. To je primitivna spremenljivka npr. celoštevilčna, realna, digitalna, spominsko sporočilo ali znak.

Razredi definirajo vrste metod, ki se nanašajo na objekte določenega razreda. Metode določijo obnašanje razredov. Vsaka metoda ima ime, ničlo oz. več argumentov, kjer ima vsak argument svoj tip spremenljivke. V nekaterih modelih so polja skrita za metodami tako, da vsak dostop do branja ali nastavljanja polja poteka preko iste metode.

⁶ Dan R. Olsen, Jr, Developing User Interfaces



Slika 11: Gumbi za navigacijo po delih procesa

Za nazornejši prikaz komunikacijske kompleksnosti v uporabniškem vmesniku uporabimo sledeči primer; na sliki 11 vidimo gube za navigacijo po določenih delih procesa. Vsak gumb se obnaša v skladu z našimi pričakovanji, ker etiketa na gumbu vnaprej opiše akcijo. Te gube lahko na osnovi medsebojne podobnosti oz. funkcionalnosti razdelimo v razrede. Izbor določenega gumba iz razreda izvrši akcijo, ki je vidna na zaslonu.

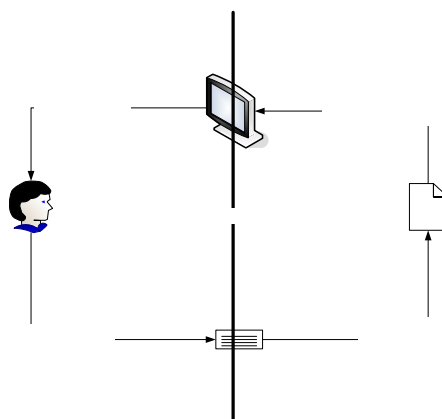
Najpomembnejši koncept objektov in razredov je, da vsak objekt nosi s seboj celotno definicijo svojega obnašanja. To omogoča programu, da komunicira z objektom, medtem, ko ignorira ostale.⁷

4 PREGLED SKOZI UPORABNIŠKI VMESNIK

4.1 Interaktivni cikel

Ko se na zaslonu pojavi aplikacija mora uporabnik razumeti, kaj je prikazano, ovrednotiti prikazano in narediti določene spremembe modela (slika 12). Ko uporabnik določi zastavljen cilj, izbere dejavnosti in tvori vhode, ki temeljijo na tej dejavnosti. Programska aplikacija mora nato prevesti vhode in tvoriti spremembe v notranjem podatkovnem modelu, ki ga program vzdržuje. Uporabnikova sprememba vhodne kode v podatkovnem modelu povzroči spremembe na celotnem oz. določenem delu zaslona, ki je krmiljen s tem modelom. Cikel se nato ponavlja.

⁷ Dan R. Olsen, Jr, Developing User Interfaces



Vrednoti in razume
zapis na zaslonu

Slika 12: Interaktivni cikel

Zaslon

Sklepamo lahko, da je primarna funkcija uporabniških vmesnikov manipulacija z informacijami.

Komunikacija ČLOVEK /

Don Norman, računalniški analitik, je postavil dve tezi t.i. brezni^{Uporabnik} v uporabnikovem razumevanju, ki lahko prekineta interaktivni cikel. Ti dve tezi sta »brezno vrednotenja« ter »brezno izvršitve«.

Izbira cilje, akcije
tvori vhod

Brezno vrednotenja

Vhodna er

»Brezno vrednotenja« se nanaša na dogodke uporabnikovih pogledov na zaslon, njihove interpretacije ter vrednotenja vidnega glede na njihove končne cilje. Vrednotenje bi lahko definirali kot vizualno in miselno stanje uporabnikov, ki morajo slediti predhodnim akcijam, postaviti morajo podlago za odločitve ter razmisliti o poznejših ukrepih.

Vrednotenje je lahko zaradi različnih dejavnikov neuspešno, nastane »brezno vrednotenja«. Ergonomski faktorji lahko povzročijo probleme vrednotenja. Na primer, tekst je težko berljiv, pomembna informacija je predstavljena z nizko kontrastnimi barvami, tako da jo uporabniki spregledajo.

Problemi, ki nastanejo pri načrtovanju in videzu in tudi lahko povzročijo probleme vrednotenja so:

- Točke so neučinkovito grupirane, tako da uporabniki ne vidijo pomembnih povezav.
- Pomembna informacija je lahko nepravilno postavljena, tako da uporabnik pri hitrem pregledu okna spregleda pomembno informacijo.
- Informacije se lahko pojavljajo hkrati brez poudarkov ali osvetljevanj, zaradi tega se mora uporabnik posebej potruditi, da najde želeno informacijo.

Eden izmed razlogov za nastanek »brezna vrednotenja« je pravočasnost. Ni zadosti, da je informacija na zaslonu prikazana hitro oz. v realnem času. Mora biti tudi razumljiva tako, da je omogočeno vrednotiti informacijo brez dolgotrajnih analiz. Enostavni objekti oz. informacije morajo biti očitni.

Brezno izvršitve

Drugi problem uporabnikovega razumevanja je »brezno izvršitve«. To brezno je izraženo, ko uporabnik uporabniškega vmesnika razume videno na zaslonu, ne ve pa, katero zaporedje vhodnih zahtev bo omogočalo doseganje končnega cilja. Veliko je faktorjev, ki povzročajo ta tip brezna:

Kot prvega lahko omenimo pomanjkanje razumevanja uporabnika, s katerimi podatki, akcijami in interakcijami lahko manipulira. Za uporabnika, ki ni nikoli uporabljal računalnika, je relacija med miško in objektom na zaslonu nejasna. Ravno tako navigacijske ikone nimajo nobenega pomena za uporabnika, ki prvič vidi računalnik. Slike na ikonah niso zadostni pokazatelj uporabniku, da bi le ta premaknil miškin kazalec na ikono, pritisnil miškin gumb in izvedel enako akcijo kot jo v realnem svetu. Ena izmed rešitev problema je skladno vmesniško načrtovanje, ki pomaga uporabniku jasno razločiti, kateri vhodi bodo imeli efekt na vizualne objekte. Seveda tukaj ne smemo pozabiti nekaj dnevnega raziskovalnega treninga uporabnika po uporabniškem vmesniku.

Drugi razlog je nezadostna povratna informacija odziva vhoda. Poznamo primere, kjer uporabnik tvori izbrane vhode, povratne informacije, ki pove, da je uporabnik bližje

svojemu cilju, na zaslonu pa ni. To vsekakor pelje v zmedo uporabnika in uporabniški vmesnik. Eno od načel direktne manipulacije je, da je uporabniku omogočeno varno raziskovanje vmesnika. To načelo pomaga premagati »brezno vrednotenja«, tako da je uporabniku omogočeno testirati možne akcije.

Modelna učinkovitost

Zadnji problem, ki bi ga bilo potrebno omeniti, je modelna učinkovitost naprave, ki jo program nadzoruje. Na tej točki je potrebno povedati, da se modeli strojev razlikujejo od kupca do kupca, saj imajo le ti različne zahteve za procesno linijo, pa čeprav stroji opravljajo enako delo v procesni liniji. Tako kot stroji se razlikuje tudi programska oprema in posledično tudi uporabniški vmesniki. To pomeni, da uporabnik oz. operater ne bo mogel izvajati postopkov v enakem zaporedju pri različnih kupcih. Vendar se tega problema navadno uporabniki oz. operaterji ne zavedajo. Ne zavedajo pa se tudi tega, da stroji zahtevajo zaradi svoje različne notranje gradnje svojevrstno programsko opremo, ki zahteva drugačne postopke akcij.

4.2 Interaktivno okno

Pomembna karakteristika grafičnih aplikacij je postavljanje interakcije skozi relativno ozek vidik pogleda na svet. Tako postavljanje se imenuje interaktivno okno. V zvezi s tem lahko govorimo o omejitvah glede računalniških zaslonov ter človeške pozornosti.

Omejitve glede računalniških zaslonov

Za uporabnike oz. načrtovalce, ki so uporabljali DOS in PC world je bilo interaktivno okno veliko približno 640×480 slikovnih točk, lahko je bilo tudi manjše. Verjetno bi bila lahko ločljivost zaslona večja (fizično je bilo mogoče upravljati tudi z večjimi resolucijami računalniškega zaslona), ampak zgleda, da je bil takšen ozek pogled na interaktivni svet nekako zakoreninjen v kulturo programiranja in uporabe.



Slika 13: Primerjava velikosti resolucij na zaslonu

Današnji interaktivni svet nima več tako velikih fizičnih in programskih omejitev glede ločljivosti zaslonov (slika 13). Naš uporabniški vmesnik je velikosti 1024×768, videli pa smo že vmesnike, ki uporabljajo od 1200×1200 slikovnih točk naprej. Zanimivi so tudi sistemi, ki uporabljajo več zaslonov povezanih v celoto. Ta način vezave zaslonov omogoča zelo veliko delovnega prostora, kar je uporabno za aplikacije, ki uporabljajo veliko zaslonskega prostora. Načrtovanje interaktivnega okna je dobrodošlo za reduciranje brezna vrednotenja in brezna izvršitve.

Omejena človeška pozornost

Najpomembnejši dejavnik načrtovanja učinkovitega informacijskega okna je pritegnitev pozornosti uporabnika. Kot lep primer lahko iz našega projekta vzamemo procesne strani. Tukaj smo vse stroje tako na vhodu kot na izhodu animirali s pomočjo različnih barv, ki pritegnejo pozornost uporabnika oz. operaterja. Tako so npr. stroji v normalnem delovanju obarvani zeleno, ob napaki pa rdeče. Enako lahko govorimo o alarmih, ki so obarvani z različnimi barvami (glede na prioriteto alarma) ter alarmni vrstici, ki je v nogi SCADA aplikacije na vseh procesnih straneh, kjer opozarja oz. pritegne pozornost uporabnika (operaterja) na nepravilnosti v delovanju proizvodne linije.

4.3 Proces načrtovanja vmesnika

V zvezi s prejšnjimi poglavji lahko sedaj preidemo k procesu načrtovanja uporabniškega vmesnika. Proces načrtovanja poteka na sledeči način:

- analiza nalog,
- funkcionalnega modela,
- predstavitev,
- interaktivnega okna in
- akcij.

Pomembna točka, ki potrebuje pozornost je še spoznanje, da načrtovanje ni linearni proces. Uporabniškega vmesnika ne začnemo graditi na začetku in nato se natančno držati premočrtnega načrtovanja. Načrtovanje je ponavljajoči se krožni cikel, kot je prikazano na sliki 9. Iz tega izhaja spoznanje, da je med gradnjo uporabniškega vmesnika potrebno dodatno vrednotenje vsakega koraka, kjer naletimo na problem. Odločitev in izvedba mora biti sprejeta ne glede na to, v kateri fazi gradnje pride do napake.

Analiza nalog

Analiza nalog je prvi in verjetno tudi najpomembnejši korak pri gradnji uporabniških vmesnikov. Pomembno je osredotočiti na naloge, ki jih hočejo uporabniki oblikovati. Če sistem ne služi oblikovanju uporabniških nalog, potem je tak sistem neuporaben, kar posledično pripelje do tega, da uporabnik sistema ne kupi. Cilj tega podpoglavja je razumevanje uporabnikovih zahtev, tehnik, informacij in strategij za doseg zastavljenega cilja.

Povedati je potrebno, da je izhod tega podpoglavja spisek nalog. Ta spisek predstavlja opravila uporabnikovih potreb, ki jih mora nov sistem opraviti. Spisek navadno vsebuje opise informacij, ki jih uporabnik uporablja, standardno terminologijo in simbole ter

opise splošnega načina izvedbe nalog. Ne smemo pozabiti informacije o programski opremi, s katero je uporabnik domač oz. jo pozna.

Načrtovanje funkcionalnega modela

Poznavanje uporabnikovih zahtev nas pripelje do definiranja aplikacijskega modela, ki bo pospešil doseganje ciljev oz. zahtev. Funkcionalni model pove česa vsega je sistem zmožen, vendar ne vključuje podrobnosti kot so npr. vezanje tipk ali grafični zgled strani.

Pri načrtovanju funkcionalnega modela pridemo navzkriž z načrtovanjem vizualne podobe. Uporabniki lahko hitro razumejo enostavne vizualne podobe. Problem pa nastane pri velikih sistemih, kjer mora funkcionalnost zadostiti različnim potrebam tehnologije in produkta. Podrobno in pazljivo načrtovanje sistemskih funkcij lahko dovoljuje širok spekter poti za doseg določene funkcije. Pri tem lahko naletimo na problem nerazumevanja uporabnika, saj le ta lahko misli, da so poti enake funkcijam, ki jih sistem izvaja, kar pa ni res. Funkcionalni model je uporaben vmesnik osredotočanja med uporabnikom ter gradnjo.

Funkcionalni model v glavnem sestavlja razrede objektov, ki so predstavljeni v sistemu. Razredi so razdeljeni na nad-razrede in pod-razrede, ne samo za izgradnjo, temveč tudi za uporabnika, ki mora na abstrakten način razumeti, kako se model obnaša. Pomemben del pri načrtovanju funkcionalnega modela je predstavitev informacije pred manipulacijo (kako, na kakšen način). Splošno je znano, da je predhodno informiranje za uporabnika boljše kot direktna manipulacija s sistemom. To mora biti v funkcionalnem modelu jasno opredeljeno.

Poenostavljanje funkcionalnega modela ne sme izpustiti nalog, ki jih mora uporabnik zaključiti, poleg tega pa ne sme pustiti uporabniku preveč svobode pri izbiranju nalog, saj taka aplikacija postane hitro nepregledna in neučinkovita.

Da se izognemo težavam, je potrebno ponavljajoče vrednotiti funkcionalni model z analizami nalog za uporabnika. Preveč podrobnih nalog lahko povzroči zmedo. Samo

ponavljajoče vrednotenje znanja uporabnika o lastnih potrebah bo rešilo to točko funkcionalnega modela.

Načrtovanje predstavitev

Po pridobitvi funkcionalnega modela z vsemi potrebnimi informacijami, s katerimi pride uporabnik v stik, je potrebno načrtovati vizualno predstavitev izbranega modela. Vizualna predstavitev informacij je pomembna zaradi zmanjševanja »brezna vrednotenja«. Vizualna predstavitev ukrepov, ki jih uporabnik lahko izbere, je pomembna za zmanjševanje »brezna izvršitve«. Tako mora biti vizualno jasno, katere parametre lahko uporabnik spreminja in kateri so prikazani kot enostavna informacija.

Za dobro predstavitev je potrebno vedeti nekaj o uporabnikovem dojemanju slik. Ljudje na primer zelo težko berejo modre črke na rdeči podlagi ali pa gledajo majhne modre objekte na rdeči podlagi. Ravno tako imajo različne velikosti in tipi pisav različen učinek na branje znakov. V spodnjih tabelah vidimo dobre in slabe barvne kombinacije, za vzbujanje človeške pozornosti (tabeli 1, 2)

Tabela 1: Dobre barvne kombinacije

OZADJE	TANKE ČRTE	TEKST	DEBELE ČRTE	POLNJENA POLJA
bela	modra	črna	črna	modra
črna	bela	rumena	rumena	bela
rdeča	rumena	bela, črna	črna	rumena, bela
zelena	črna	modra	črna	rdeča
modra	bela	rumena	rumena	vijolična
modro zelena	modra	črna, rdeča	modra	črna
vijolična	črna	bela	modra	črna
rumena	modra	rdeča	modra	črna

Tabela 2: Slabe barvne kombinacije

OZADJE	TANKE ČRTE	TEKST	DEBELE ČRTE	POLNJENA POLJA
bela	rumena	modro zelena	rumena	modro zelena
črna	modra	rdeča	modra	vijolična
rdeča	vijolična	modra	vijolična	modra, zelena
zelena	modro zelena	vijolična, rumena	modro zelena	vijolična, rumena
modra	zelena	rdeča, črna	zelena	rdeča, črna
modro zelena	zelena	rumena	rumena	zelena, rdeča
vijolična	zelena	rdeča, modro zelena	modro zelena	zelena, rdeča
rumena	bela	modro zelena	bela	modro zelena

Pri načrtovanju vizualne predstavitve je potrebno omeniti tudi programsko arhitekturo, ki mora v realnem času osveževati zaslon glede na spremembe aplikacijskega podatkovnega modela.

Načrtovanje interaktivnega okna

Načrtovanje interaktivnega okna je pomembno zaradi prilagajanja umestitve informacij na zaslon. Informacija lahko leži na dvo-dimenzionalni ravnini, okno pa se premika po tej ravnini tako, da se ujema z zaslonom. Na tak način so narejeni urejevalniki besedil ter orodja za risanje.

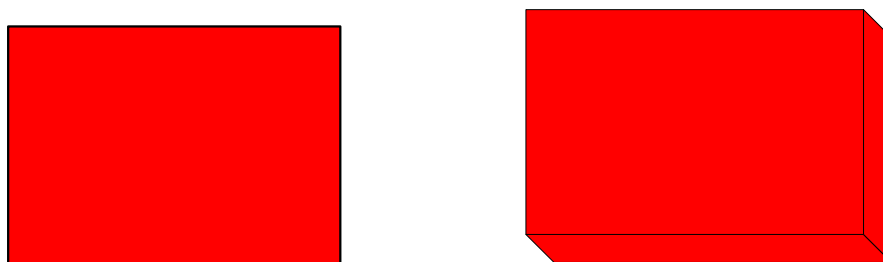
Interaktivno okno se ne nanaša samo na prikazovanje informacij, ampak tudi na vizualno predstavitev možnih akcij. Če je število možnih akcij preveliko, da bi bile vse akcije dostopne iz določenega okna, lahko uporabimo druge prijeme za predstavitve ukrepov. Tako se npr. potrjevanje akcij organizira s pojavnimi okni (ang. Pop-up,), ki se pojavijo ter delno tudi prekrijejo obstoječe okno. Na ta način lahko zadostimo dvema potrebama načrtovanja uporabniškega vmesnika in sicer: vzbudili smo pozornost uporabnika ter rešili možen problem pomanjkanja okenskega prostora pri potrjevanju akcij.

Ena izmed prednosti dobre direktne manipulacije z uporabniškim vmesnikom je vidnost objektov na zaslonu, s katerimi lahko manipuliramo. Tukaj pa se lahko tudi zaplete, saj nevidnost objektov iz katerega koli razloga onemogoča akcije. Če uporabnik objekta ne vidi, z njim ne manipulira.

Načrtovanje interaktivnega okna mora upoštevati strukturo nalog in akcij. Le te morajo biti razdeljene na sorodne množice tako, da je predstavitev organizirana na način zagotavljanja potrebne informacije, v sorodu s preostalimi nalogami in akcijami. Ta način organiziranja preprečuje uporabniku hkratno manipulacijo s podobnimi nalogami in akcijami v uporabniškem vmesniku, saj mora uporabnik za doseg končnega cilja iti skozi množico soodvisnih nalog in akcij. Struktura organiziranosti mora biti podobna kot v resničnem svetu.

Načrtovanje akcij

Po uspešno načrtovani predstavitvi in postavitvi informacij na zaslon je sedaj čas za načrtovanje tehnik, ki omogočajo uporabniku manipulacijo z objekti. Ta razvojna faza je redko ločena od ostalih faz razvoja in je pogosto usklajena z razvojem predstavitve ter organizacije uporabniškega vmesnika. Eden od razlogov za skladen razvoj je podan na sliki 14. Tukaj lahko vidimo dve enaki sliki. Ena zglada kot statična predstavitev informacije, druga pa ima pojavno obliko fizičnega gumba, na katerega uporabnik lahko pritisne. Tak način predstavitve gumba je učinkovit za zmanjševanje brezna izvršitve. Taka vizualna povabila se imenujejo »privoščila«. Uporaba standardnih »privoščil« je v veliko pomoč uporabnikom, ki že na prvi pogled ve, kaj je predstavitev informacije in kaj je akcija v uporabniškem vmesniku.⁸



Slika 14: Primer predstavitve akcije

⁸ Dan R. Olsen, Jr, Developing User Interfaces

5 SCADA PROGRAMSKO ORODJE

SCADA je programsko orodje, ki se uporablja za razvoj in izvedbo uporabniškega vmesnika (SCADA aplikacije) na računalniških sistemih. Kratica SCADA v angleškem jeziku pomeni Supervisory Control And Data Acquisition oz. nadzor in pridobivanje podatkov. Taki tipi programskega orodja so se najprej uporabljali v velikih računalniških centrih (računalniki so bili izredno veliki), ki so nadzorovali večje proizvodne sisteme npr. elektrarne, valjarne ipd. Z upadanjem cen, višanjem moči, novimi možnostmi komunikacije in zmanjševanjem velikosti računalnika, so SCADA aplikacije postale uporabne tudi za širšo uporabo. Tako jih sedaj najdemo v porazdeljenih računalniških sistemih vodenja, ki so sestavljeni predvsem iz miniračunalnikov, razvoj komunikacij pa je poskrbel za nadzor procesa iz oddaljenega delovnega mesta.

Današnji operacijski sistemi, na katerih se uporablja SCADA:

- Microsoftovi operacijski sistemi pokrivajo 90% trga (NT, Win 95, Win 98, Win 3.11, DOS);
- UNIX, QNX, OS2 pod 10%

Na tržišču najdemo različne proizvajalce SCADA sistemov, med katere štejemo:

- Intellution (IFIX),
- Wonderware (InTouch),
- USDATA (Factory Link),
- Siemens (WinCC),
- Rockwell Software (RSView)
- ipd.

Osnovne funkcije nadzornih sistemov:

- zbiranje procesnih podatkov;
- prikazi in ročno vodenje;
- alarmiranje;
- shranjevanje procesnih podatkov;
- tvorjenje poročil.⁹

Zbiranje procesnih podatkov

Če hočemo celovito obvladovati industrijski proces, moramo sprotno spremljati proces v realnem času. Zaradi tega morajo orodja SCADA vsebovati naslednje arhitekture:

- izvajanje v realnem času,
- vmesnike za dostop do procesnih podatkov.

Aplikacije SCADA so sposobne brati, pisati podatke iz procesnega računalnika, prikazovati okna na ekranu in shranjevati na disk po vnaprej določenih pravilih. Tako lahko podatke zbiramo v določenem časovnem intervalu ali dogodkovno.

Prikazi in ročno vodenje

Poleg funkcije zbiranja procesnih podatkov morajo aplikacije tipa SCADA omogočiti operaterju sistema neposreden vpogled in poseg v delovanje procesa. Povezava med operaterjem in SCADO mora biti čim bolj enostavna in razumljiva. Podatki so lahko podani v grafični ali tekstovni obliki.

Alarmiranje

Alarmiranje je zelo pomembna funkcija nadzornih sistemov, saj le ta obvešča operaterja o določenih odstopanjih merjenih količin. Aplikacije SCADA imajo možnost definiranja

⁹ Medmrežje, Fakulteta za elektrotehniko, računalništvo in informatiko Maribor

dogodkov glede na njihovo kritičnost ter jih na podlagi tega tudi grupirati (alarmi in sporočila).

Arhiviranje podatkov

Dokaj pomembna funkcija je tudi možnost arhiviranja podatkov v podatkovno bazo. Orodja SCADA so dandanes zmožna že prenosa podatkov v standardne baze kot npr. Microsoft Access, SQL Server, Oracle ipd.

Tvorjenje poročil

Orodja SCADA so sposobna tvoriti različne tipe izpisov poročil na osnovi podatkov iz procesa.

Lastnosti programskih orodij SCADA

Lastnosti teh orodij gledamo iz dveh vidikov:

- z vidika razvijalca programske opreme, katerega naloga je podati uporabniku prijazno okolje in
- z vidika uporabnika oz. operaterja, katerega zahteve se nanašajo predvsem na funkcionalnost aplikacije.

Pri sodobnih SCADA sistemih lahko govorimo o naslednjih lastnostih:

- uporaba objektno orientirane paradigme;
- modularna zgradba;
- integriran sistem varnosti in zaščite;
- povezljivost in odprtost ter standardnost¹⁰;

¹⁰ S. Strmčnik, Celostni pristop k računalniškemu vodenju procesov

6 PRIMER IZDELAVE SCADA APLIKACIJE

V tem poglavju se bomo osredotočili na izdelavo SCADA aplikacije in poizkušali kar čimbolj nazorno prikazati njeno delovanje in izgradnjo. Za boljše razumevanje bralca se bomo najprej na kratko osredotočili na valjanje in tipe valjarskih strojev, nato bomo pregledali stroje v valjarski progi Federal Mogul, saj je zgled in izgradnja tega tipa aplikacije SCADA odvisna od postavitve in različice valjarske linije oz. proge. Naslednja, pa bodo govorila o programskem orodju InTouch, v katerem smo aplikacijo SCADA gradili ter podrobno opisali izgradnjo in delovanje SCADA aplikacije oz. t.i. grafičnega uporabniškega vmesnika za nadzor valjarske proge.

6.1 Valjanje

Pri valjanju gre za proces plastične deformacije, ki ga uporabljamo za izdelavo polizdelkov, kot so različni tipi pločevine, trakov, profilov, ipd. V uporabi danes srečamo različne tipe valjnih valjev npr. gladke (valjanje polčevine) in kalibrirne valje (valjanje profilov). Profile valjamo z vročim, pločevino tudi s hladnim valjanjem.

Valjanje spada med najvažnejše postopke za izdelavo polizdelkov, saj dandanes predelamo med 80% in 95% vsega jekla v valjarnah. Valjamo pa tudi že zelo dolgo, saj je bila prva valjarna ustanovljena že leta 1773 v Nemčiji, leta 1820 pa tudi že v Angliji za potrebe železniškega prometa. Valjanje se je prijelo v industriji jeklarskih polizdelkov tudi zaradi širokega spektra valjanih materialov. Valjamo lahko čiste kovine kakor tudi zlitine, ki so primerne za plastično deformacijo, saj se postopek ne razlikuje glede na tip valjanca.

Postopek valjanja

Valjanje se izvaja preko dveh ali več valjev, ki se vrtijo v nasprotnih smereh. S tem dosežemo povečanje dolžine in zmanjševanje prereza. Zaradi tega morajo imeti valji manjši medsebojni razkorak, kot je prerez valjanca. Valjanje se prične takrat, ko valjanec pride v stik z valji, ki ga deformirajo (plastična deformacija) v željeno obliko.

Tukaj bi se radi omejili na valjanje jekla, saj bomo v nadaljevanju opisovali predvsem postopke, ki se nanašajo na njegovo proizvodnjo.

Vsako jeklo se ne valja enako dobro. Valjanje je odvisno predvsem od stopnje ogljika v valjancu. Tako se na primer jekla, ki vsebujejo do 0.8% ogljika, valjajo zelo dobro, pri jeklih, ki imajo višjo raven ogljika v kemični sestavi, pa se uporablja poseben postopek toplotne obdelave z nižjo hitrostjo valjanja. Dobro je vedeti, da je sposobnost jekla za plastično deformacijo odvisna predvsem od sledečega:

- sestave,
- načina in velikosti preoblikovanja,
- temperature in
- hitrosti valjanja.

Na splošno je valjarna oz. valjarska proga sestavljena iz grupe različnih strojev, ki morajo biti vezani v točno določenem zaporedju in legi glede na tip polproizvoda. Vsaka valjarna mora imeti tudi pomožne naprave, ki dopolnjujejo proizvodni proces, saj bi le brez teh proizvodnja oz. valjanje potekalo izredno počasi in nezanesljivo. Zaradi tega te naprave štejemo k tehnološkemu postopku. Te naprave so različne glede na tip valjarne, lahko pa naštejemo nekaj naslednjih primerov le teh:

- peči za segrevanje in žarjejnje valjanca,
- razni razprsilci za čiščenje,
- hladilni stolpi za hlajenje,
- škarje za rezanje,
- privzdigovalne mize,
- razni žerjavi in valjčni transporterji za premikanje valjancev pred ali po valjanju.

Valjarske stroje ločimo po tem, koliko valjev posamezni stroj vsebuje.

Duo valjarski stroj, ki je sestavljen iz dveh valjev, ki se vrtita v nasprotni smeri, razmak med valjema in s tem naravnano debelino valjanca določa operater linije poljubno, s pomočjo hidravličnih batov, ki delujejo neodvisno eden od drugega, njegova delovna smer je le ena. To je lahko tudi pomankljivost teh tipov strojev, saj mora valjanec potovati nazaj na vhod stroja, ali pa mora biti valjarska linija sestavljena iz zaporedja enakih strojev, kar pa posledično viša stroške in čas proizvodnje polizdelkov. Pomankljivosti Duo Valjarskih strojev odpravlja **reverzibilni duo** valjarski stroj, ki je sposoben valjati v dveh smereh, mora pa biti zelo robusten, saj je izpostavljen neprestanemu spreminjanju smeri vrtenja valjev. **Trio valjarski stroj** je sestavljen iz treh valjev, kjer se krajna vrtita v isto smer, sredinski pa obratno. V tem primeru je sredinski valj fiksiran, medtem ko se krajna valja lahko prosto premikata v vertikalni oz. horizontalni (odvisno od izvedbe trio valjarskega stroja) smeri. Pri vseh strojih tega tipa linija potrebuje posebne dvigalne mize za dviganje oz. spuščanje valjanca. Postopek poteka na sledeči način:

- valjanje na spodnji strani sredinskega valja,
- valjanje na zgornji strani sredinskega valja,
- dvigovanje valjanca na višji nivo,
- spuščanje na nižji nivo.

Postopek se ponavlja tako dolgo, dokler valjanec ne doseže zelenih lastnosti in oblike. Slaba lastnost teh strojev je veliko mehanskega premikanja valjancev, kot dobro pa bi lahko šteli možnost valjanja dveh ali več valjancev hkrati. **Kvarto valjarski stroj** je sestavljen iz štirih valjev (preseki so parno različni), ki so postavljeni vertikalno. Ta stroj uporabljamo za valjanje trakov in pločevine, lahko tudi v hladnem stanju. Stroj s tako zgradbo, kot jo ima kvarto valjarski stroj, omogoča natančnejše valjanje, saj zunanja debelejša valja nista v neposrednem stiku z valjancem, sta pa v stiku z delovnima valjema in prenašata nase upogibne napetosti in posledično povzročata trde in toge delovne valje. Za večjo natančnost valjanja poznamo valjarske stroje tudi s 6 ali 12 valji.

6.1.1 Opis proizvodne valjarske linije Federal Mogul po valjarskih strojih

Valjarska linija Federal Mogul je sestavljena iz treh glavnih delov, in sicer:

- Vhodni del valjarske linije:
 - odvijalnik traku,
 - valji za stiskanje z ravnalnimi valji,
 - izravnalni valji in
 - akumulator jeklenega traku.
- Procesni del,
- izhodni del:
 - kvarto valjarski stroj (mlin) in
 - navijalnik traku.

Kot je vidno je procesni del valjarske proge neznanka, saj je avtomatizacija ter posledicno SCADA aplikacija namenjena samo vhodnemu ter izhodnemu delu valjarske proge.

6.1.1.1 Vhod valjarske linije

Odvijalnik traku in njegove sestavne enote

Odvijalnik traku slika 15, je na začetku valjarske proge oz valjarne. Takoj po namestitvi jeklenega zavitka nam odvijalnik pomaga pri odvijanju in točni namestitvi jeklenega traku med obdelovalne valje sosednjega valjarskega stroja.

Celoten odvijalnik jeklenega traku je zmožen pravokotnega premika glede na smer potovanja traku. Ta premik mu omogoča centriranje jeklenega traku. Centriranje se

izvaja v mejah $\pm 150\text{mm}$ v levo oz. desno stran. Sistem pravokotnega premikanja odvijalnika je krmiljen s pomočjo krmilnega paketa.

Roka za držanje glave jeklenega traku (na sliki 15) pomaga pri namestitvi traku med obdelovalne valje sosednjega stroja, saj usmerja glavo traku neposredno proti sosednjemu stroju. Roka je krmiljena s pomočjo hidravličnega bata, ki omogoča spuščanje in dvigovanje. Roka se spušča oziroma dviga ročno (preko operaterja) ali pa kot del avtomatske sekvence. Med delovanjem v področju »napetosti« je roka praviloma dvignjena.

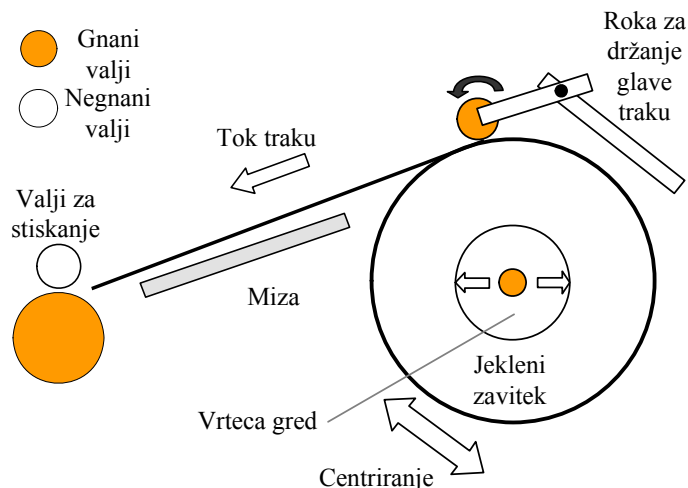
Miza na sliki 15 se uporablja za pomoč pri nameščanju glave traku v sosednji stroj. Glava mize se je zmožna preko vhodnih podatkov o preseku jeklenega zavitka postaviti v pravilen položaj, s tem da se zavrti za določen kot ter iztegne za določeno dolžino. Po končani namestitvi jeklenega traku v sosednji stroj se miza vrne v izhodiščni položaj (je umaknjena).

Vrteča se gred odvijalnika na sliki 15 z večanjem svojega preseka klini (poveča trenje med gredjo in jeklenim zavitkom) jekleni zavitek od znotraj ter s tem onemogoča drsenje gredi po notranji strani jeklenega zavitka. Na koncu odvijalnega procesa pa se gred stisne, da prepreči prekomerno razširjanje, saj je takrat jekleni zavitek izredno lahek.

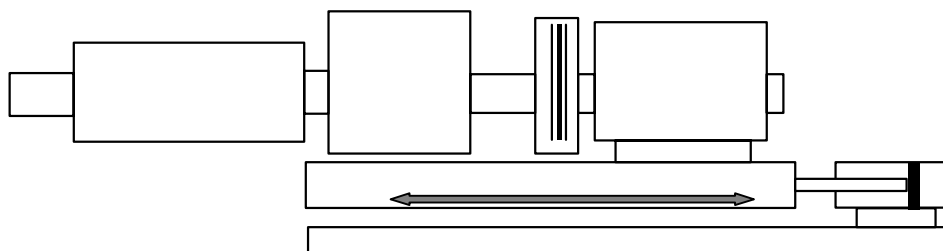
Ko je trak nameščen v sosednji stroj, odvijalnik preide iz počasnega teka (ang. jog) v delovanje, imenovano »napetost«. V tem stanju delovanja odvijalnik povzroča povratno napetost na jekleni trak zaradi spreminjajočega se preseka jeklenega zavitka. Napetost odvijalnik ustvarja z AC motorjem (hkrati je to tudi glavni pogon odvijalnika), ki je preko mehanike prestav vezan na vrtečo se gred. Gred se pomika z izredno majhnimi premiki in hitrostjo.

Motor napaja hidravliko (slika 16), ki maže mehaniko prestav. Mazalne črpalke dobijo olje iz posebnega oljnega rezervoarja, ki se ravno tako kot motor in črpalke je v posebnem zaprtem prostoru na odvijalnik. Celota se imenuje krmilni motor. Odvijalnik

ima seveda tudi zavore, ki ustavi odvijanje traku. Zavore morajo biti na vseh strojih v liniji medsebojno povezane, saj le to zahteva varnostni standard.



Slika 15: Shematski prikaz odvijalnika

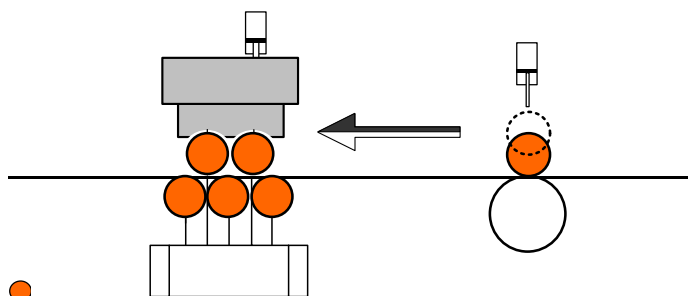


Slika 16: Shematski prikaz (stranski ris) odvijalnika

Valji za stiskanje z ravnalnimi valji

Vhodni stiskalni valj je sestavljen iz dveh valjev, ki pomagata pri nameščanju glave traku v ravnalne valje. Spodnji valj je pritrjen, zgornji pa se lahko premika po vertikali s pomočjo hidravličnega cilindra. Krmiljen je s PLK-jem, preko SCADA aplikacije, kjer nastavljamo parametre vertikalnega pomika. Zgornji valj se premika samo v času lovljenja glave traku (slo. prosti tek, ang. jog), kasneje pa se spusti vse do jeklenega traku.

Ravnalne valje se uporabljajo za odstranjevanje težjih materialov iz traku na glavi in repu vsakega jeklenega zavitka. Omogoča tudi kasnejše lažje hranjenje linije, saj je trak, ki izhaja iz ravnalnih valjev, dokaj raven. Slika 17 prikazuje shematski prikaz stiskalnih in ravnalnih valjev.



Slika 17: Stisljivi in ravnalni valji

Pet trdih jeklenih valjev se vrte v mazalnih podpornikih, ki so podprti v glavnem ohišju ravnalnih valjev.

Trije spodnji valji so fiksirani, zgornja dva pa sta vezana na ločene podporne bloke, ki lahko neodvisno spreminjajo lego (vertikalno) s pomočjo hidravličnih škripcev. Te škripce napaja hidravlični motor. Ravno tako kot zgornji stisljivi valj tudi premična ravnalna valja preideta v spodnjo skrajno mejo, ko glava traku preide skozi izhodni valj. To pomeni, da so zgornji valji odprti v sekvenci lovljenja glave traku oz. počasnem teku (ang. jog), drugače so zaprti. Krmiljeni na enak način kot stiskalni valji.

-Ravnalni valji:

Izravnalni valji

Gnani valji

Mehanika prestav

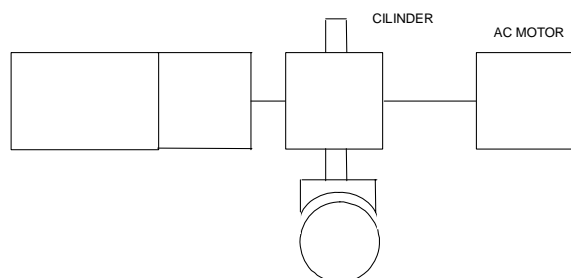
Izravnalni valji so v liniji takoj za valjarskim strojem, katerega funkcionalnost je neznana, saj ta stroj izdelujejo in avtomatizirajo v Nemčiji.

Izravnalni valjarski stroj (slika 18) je sestavljen iz desetih zgornjih in enajstih spodnjih valjev. Vsi valji so motorizirani, možen je premik določenih valjev s pomočjo

hidravličnih cilindrov, le ti pa se napajajo s pomočjo AC motorja, ki služi tudi za pogon valjev. Tako se lahko vertikalno premika posamično pet zgornjih valjev na vhodu in izhodu stroja, spodnjih enajst valjev pa se premika sorazmerno. Premikanje valjev je kontrolirano s pomočjo SCADA aplikacije.

V času delovanja se izravnalni valji tudi obrabijo, saj je sila pritiska nanje zelo velika. Stroj ima zato tudi funkcijo hitre menjave valjev, ki se lahko izvaja med menjavo jeklenih zavitkov. Menjava se izvaja lokalno s pomočjo kontrolne mize in ni vključena v SCADA aplikacijo.

Na vhodu stroja najdemo stiskalne valje, ki služijo za pomoč pri požiranju glave traku. Z izravnalnim valjarskim strojem vplivamo na jekleni trak v smislu njegovega minimalnega podaljšanja. Podaljšanje dosežemo s kontrolo pritiska izravnalnih valjev in napetosti, ki je povzročena s pomočjo predhodnih valjarskih strojev.



Slika 18: Shematski prikaz valja v izravnalnem stroju

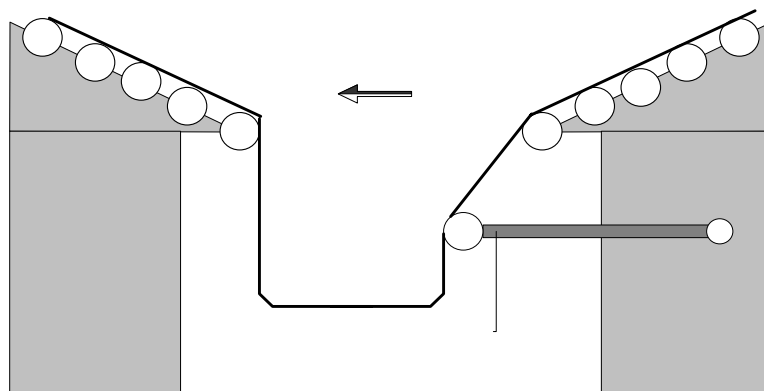
Akumulator jeklenega traku

Akumulator (slika 19) uporabljamo za shranjevanje jeklenega traku. Vanj lahko shranimo do 7 m traku. Pri valjarskih linijah najdemo akumulator predvsem zaradi razlik v hitrosti delovanja linije. Le z akumulatorjem je omogočeno nemoteno delovanje procesa s konstantno hitrostjo, saj hitrost na vhodu linije zaradi menjave jeklenega zavitka preide v ničelno.

Jekleni akumulator deluje po načelu polnjenja in praznjenja. Vhod procesa navadno obratuje z malo večjo hitrostjo kot sam proces, zato se akumulator polni. Polni se s

pomočjo akumulatorske roke, ki se sorazmerno s hitrostjo vhoda linije spušča in s tem omogoča akumulacijo. Med normalnim delovanjem je akumulator napolnjen približno 90%. Praznjenje akumulatorja se začne, ko na odvijalniku ni več jeklenega traku oz. z doseženo nastavitvijo procentualne napoljenosti v SCADA aplikaciji. Po zaznavi repa traku na odvijalniku se na akumulatorju začne počasi dvigovati akumulacijska roka, ki z enakomerno hitrostjo prazni akumulator. Če senzor na akumulatorju zazna, da preostanek jeklenega traku pade pod 15%, procesni del linije zmanjša hitrost. Pod 2,5% napoljenostjo akumulatorja pa se celotna valjarska linija ustavi, pojavi se alarm na SCADA aplikaciji.

Po zamenjavi jeklenega zavitka na odvijalniku, vhod linije pospeši nazaj na delovno hitrost. Akumulator se polni do 90% zasedenosti.



Slika 19: Shematski prikaz akumulatorja

6.1.1.2 Izhod valjarske linije

Kvarto valjarski stroj (mlin)

Mlin (slika 20) se v železarski industriji uporablja za doseganje specifičnih lastnosti jeklenega traku, ki potuje skozenj. Upravlja se lahko na dva načina:

- V produkcijskem načinu. Ta način ima dve upravljalni shemi in sicer »pritisk valjev« in »razteznost traku s sosednjimi valjarskimi strojev«. Poznamo tudi

upravljanje v ročnem ali avtomatskem načinu. Vse te funkcije lahko operater spreminja po potrebi z SCADA aplikacijo ali kontrolne mize.

- In vzdrževalnem načinu. Ta način se uporablja za menjavo delovnih in nedelovnih valjev.

Raztezek jeklenega traku skozi mlin je dosežen na dva načina: »s pritiskom valjev« ali z »razteznostjo traku s pomočjo sosednjih valjarskih strojev«.

Razteznost traku s pomočjo sosednjih strojev je najlažja pot, s katero lahko dosežemo določen raztezek. V tem primeru se valji sosednjih strojev mlina vrtijo v nasprotni smeri in s tem dosežejo, da se trak na mlinu raztegne. Za ta način raztegovanja se porabi velika količina energije.

Pritisk valjev služi kot primarna metoda za razteg. Če izberemo to metodo na operativnem oknu oz. SCADA aplikaciji, mlin regulira svojo silo na valje glede na izbrano vrednost. Zaradi tega prihaja v procesu valjanja do manjših nateznostnih sil.

Mlin za svoje delovanje uporablja delovne valje, s katerimi izvaja funkcije valjanja na trak.

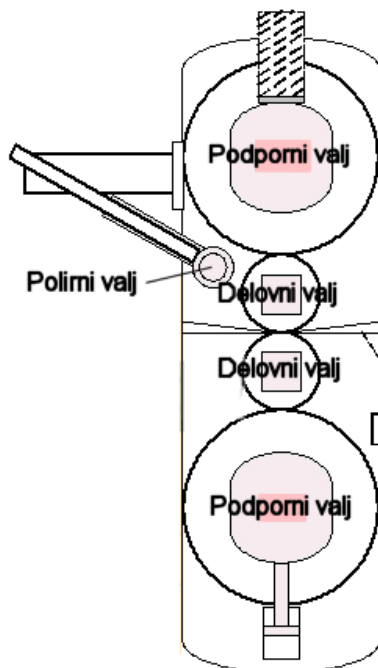
Zamenjava delovnih valjev se lahko izvaja, ko je linija nedelujoča. Ta način zamenjave omogoča uporabniku nenačrtovano vzdrževanje, kar posledično vodi tudi v nižje stroške proizvodnje.

Zamenjava podpornih valjev se lahko začne izvajati na nedelujoči liniji, konča pa se lahko tudi, ko je linija operativna.

Zamenljive valje (delovne in podporne) je potrebno programsko nastaviti (navadno to počnemo v SCADA aplikaciji) z nastavljanjem ničle, če se radij valja in debelina jeklenega traku spremeni. Tako je potrebno na novo nastaviti operativno višino valjev. Dva s pritiskom vodena klina se uporabljata za nastavljanje upogiba jeklenega traku. Delujeta tako, da s hidravlično silo vplivata na delovne valje, ki se nato prilagajata različnim debelinam jeklenih trakov.

Delovni valji se vzdržujejo s polirnimi valji, ki se vrtijo levo ali desno glede na glavo delovnih valjev. Nihajo tudi v smeri naprej nazaj tako, da lahko obkolijo celotno glavo valja. Operater lahko polirne valje upravlja ročno ali avtomatsko.

Mazanje polirnih valjev se izvaja z notranjimi razpršilnimi šobami. Šobe omogočajo čiščenje delovnih valjev na katerih se nabirajo delčki traku.



Slika 20: Shematski prikaz mlina

Navijalnik jeklenega traku

Navijalnik traku (slika 21), se nahaja na koncu valjarske proge oz. valjarne. Takoj po nastanitvi jeklenega zavitka nam navijalnik pomaga pri odvijanju in točni namestitvi jeklenega traku med obdelovalne valje sosednjega valjarskega stroja. Navijalnik je v stanju počasnega teka.

Celoten navijalnik je ravno tako kot odvijalnik jeklenega traku zmožen pravokotnega premika glede na smer potovanja traku. Ta premik mu omogoča centriranje jeklenega

traku. Centriranje se izvaja v mejah $\pm 150\text{mm}$ v levo oz. desno stran. Sistem pravokotnega premikanja odvijalnika je krmiljen s pomočjo krmilnega paketa.

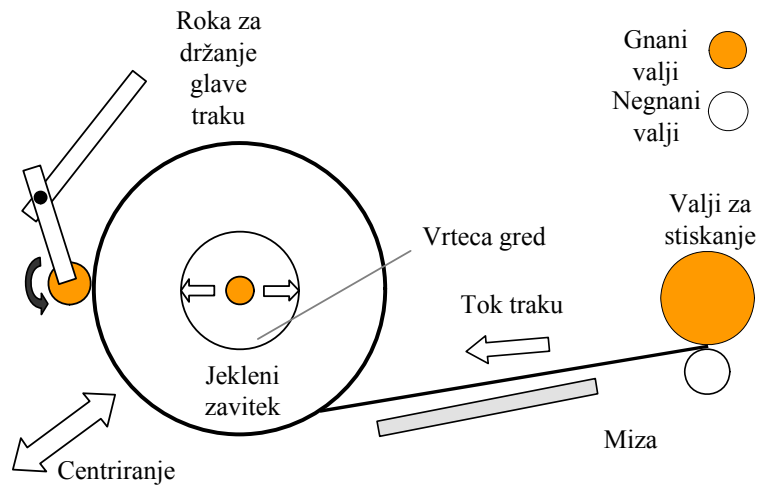
Roka za držanje glave jeklenega traku (na sliki 21) pomaga pri demontaži jeklenega zavitka (drži rep traku v pravilnem položaju). Roka je krmiljena s pomočjo hidravličnega bata, ki omogoča spuščanje in dvigovanje. Roka se spušča oziroma dviga ročno ali pa kot del avtomatske sekvence. Med delovanjem v področju »napetosti« je roka praviloma dvignjena.

Miza na sliki 21 se uporablja za pomoč pri nameščanju glave traku v navijalnik. Glava mize na osnovi vhodnih podatkov o preseku jeklenega zavitka postavi v pravilen položaj s tem, da se zavrti za določen kot. Po končani namestitvi jeklenega traku v navijalnik se miza vrne v izhodiščni položaj (je umaknjena).

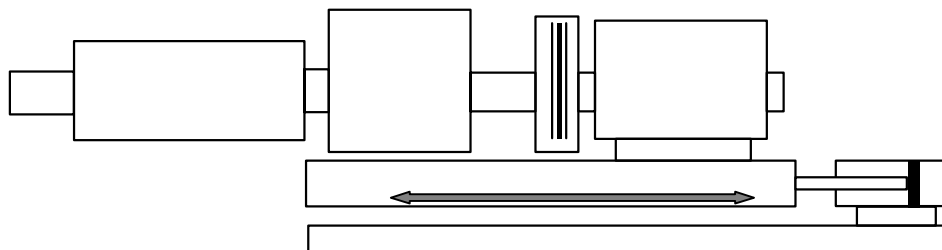
Vrteča gred navijalnika na sliki 22 z zvečanjem svojega preseka klini (poveča trenje med gredjo in jeklenim zavitkom) jekleni zavitek od znotraj ter s tem onemogoča drsenje gredi po notranji strani jeklenega zavitka. Gred vsebuje tudi posebne klešče, ki primejo glavo traku, medtem ko je le ta stisnjena. Kasneje v področju »napetosti« pa klešče popustijo, gred pa se ravno tako kot pri odvijalniku razširi.

Ko je trak nameščen, navijalnik preide v delovanje imenovano »napetost«. V tem stanju delovanja naprava tvori povratno napetost na jekleni trak zaradi spreminjajočega se preseka jeklenega zavitka. Napetost ustvarja s pomočjo AC motorja (hkrati je to tudi glavni pogon), ki je preko mehanike prestav vezan na vrtečo se gred. Gred se pomika z izredno majhnimi premiki in hitrostjo.

Motor napaja tudi hidravliko (slika 22), ki maže mehaniko prestav. Mazalne črpalke dobijo olje iz posebnega oljnega rezervoarja, ki se ravno tako kot motor in črpalke nahaja v posebnem zaprtem prostoru na navijalnika. Celotna enota se imenuje krmilni motor.



Slika 21: Shematski prikaz navijalnika



Slika 22: Shematski prikaz (stranski ris) navijalnika

6.2 Programsko orodje SCADA Wonderware InTouch

Kot primer uporabe programskega orodja SCADA bomo v naslednjem podpoglavju govorili o Wonderware programskem paketu InTouch 8.0, ki smo ga uporabljali za izgradnjo SCADA aplikacije Federal Mogul.

6.2.1 Wonderware InTouch

»Prvi produkt podjetja Wonderware je izšel leta 1989 - *InTouch za Windows*. Takrat je InTouch pomenil pravo revolucijo na tržišču, saj je kot prvi takšen produkt uporabljal operacijski sistem Windows. Kmalu se je strategija podjetja Wonderware izkazala za pravilno in danes 90 % vseh SCADA sistemov deluje v Windows okolju. Kmalu zatem je Wonderware prevzel Microsoftov protokol DDE (ang. Dynamic Data Exchange) in ga razširil za delovanje preko mreže. *NetDDE* (ang. Network Dynamic Data Exchange) je pomenil velik uspeh in Microsoft je kasneje odkupil tehnologijo in pravico uporabe NetDDE kot del svojega operacijskega sistema Windows. Prelomnico v razvoju podjetja Wonderware pomeni združitev vseh programskih orodij v enotno zbirko FactorySuite. Kar je pomenila za pisarniško okolje zbirka Microsoft Office, je pomenila za področje upravljanja industrijskih procesov zbirka FactorySuite. V zbirki je združenih več programskih orodij, ki pokrivajo praktično celotno področje računalniškega nadzora in vodenja proizvodnje (SCADA, SLC, MES, Batch).«¹¹

Programski komplet InTouch, ponuja celoten pregled nadzora, komunikacije in povezljivosti v tehnološkem procesu. Na voljo je širok spekter vhodno izhodnih strežnikov, ki dovoljujejo paketom SCADA ali ostalim Dynamic Data Exchange windows programom komunikacijo z realnim svetom PLK-ja. Notranja povezljivost tega orodja ponuja komunikacijo med operacijskimi temelji (npr. TCP/IP, DOS, OS/2 itd.) preko NetBIOS-a.

Celoten programski sklop InTouch je sestavljen iz treh glavnih delov. Prvi (WindowMaker) je namenjen gradnji grafike, animacije, oken itd. Drugi

¹¹ Medmrežje, Wonderware InTouch

(WindowViewer) prikazovanju tega, kar smo v WindowMaker-ju gradili. Tretji pa je procesna baza, ki je povezana z industrijskimi kontrolerji, vhodno/izhodnimi napravami ter Windows programi. V naslednjem podpoglavju bomo opisali primer tvorjenje procesne baze, ker je to pomemben element izgradnje SCADA aplikacije, saj le ta vsebuje vitalne podatke, preko katerih je omogočena komunikacija med SCADA orodjem (InTouch) ter PLK-jem.

6.2.1.1 Procesna baza

Procesna baza je srce InTouch-a. Pri tvorjenju procesne baze podatkov, In Touch potrebuje informacije o vhodnih nalogah (ang. tagih). Vsaki spremenljivki mora biti dodeljen tip in ime naloge.

Ime naloge je simbolno ime, ki je dodano bazi podatkov. Simboličnemu imenu naloge lahko spreminjamo parametre oz. tip (minimalne in maksimalne vrednosti, alarmiranje itd.). Naloge lahko zgradimo z uporabo treh različnih metod:

- Ročna - s to metodo lahko definiramo vsako nalogo posebej.
- Avtomatska - s to metodo nam je omogočeno takojšnje definiranje nalog s strani sistema.
- Zunanja metoda, kjer uporabljamo Wonderware-ovo koristno zunanje orodje DB Load, za prenos baze podatkov iz Excel-a v InTouch program.

V našem konkretnem primeru se za uvoz baze podatkov uporablja zunanja metoda.

Zunanja metoda

Dodajanje nalog v procesno bazo poteka preko Excelove baze podatkov (tabela 1). V njej se so podatki, ki so pomembni za komunikacijo s PLK-jem. Ti podatki so:

- naslov;

DB282- data blok, ki pove, katero zbirko podatkov uporablja SCADA (ta številka v našem primeru pove, da se pod tem naslovom skrivajo statusi naprav)

DBX – data blok boolean, maksimalno število data blokov v projektu je odvisno od PLK-ja;

010.00 – te vrednosti povejo številko besede in bita posameznega bloka.

- ime naloge;

LN_BS_SMPH_RED – za lažji pregled ima vsak naslov svoje ime;

- komentar;
- spremenljivka;
- doseg;

doseg pove, katere informacije morajo biti prikazane na SCADI (lahko so številčnega ali tekstovnega formata).

- opombe;

opombe povejo, kakšne informacije se prikažejo za določeno vrednost npr. vrednost 1 oz. »true« zahteva obarvanje z določeno barvo.

Tabela 3: Primer Excelove baze podatkov

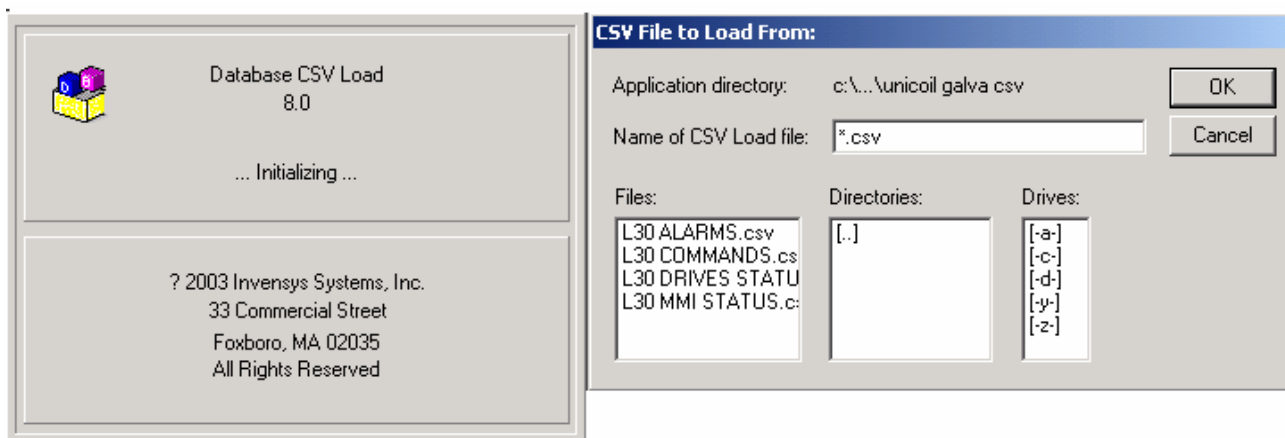
NASLOV	NALOGA	SEMAFOR	SPR.	DOSEG	OPOMBE
DBX 010.00	LN_BS_SMPH_RED	Semaphore Red Light	BB	FAULT	1=RED
DBX 010.01	LN_BS_SMPH_YELLOW	Semaphore Yellow Light	BB	ALARM	1=YELLOW
DBX 010.02	LN_BS_SMPH_GREEN	Semaphore Green Light	BB	OK	1=GREEN

DB Load

Je program (slika 23), s katerim je mogoče uvoziti Excelovo bazo podatkov v InTouch.

Bazo uvozimo na naslednji način:

- označimo, katere vrstice baze bi radi uvozili,
- zaženemo interni Excelov skupek ukazov, ki pretvori Excelove datoteke *.xls v vrednost s plavajočo vejico (ang. coma separator value) oz. *.csv (to obliko datoteke je možno uvoziti v InTouch); s skupkom ukazov tudi določimo tip komunikacije med SCADO in PLK-jem. Obstajata dve izbiri za komunikacijo: Dassidirect in S7; uporabili smo program Dassidirect;
- uporabimo DB Load za uvoz baze v InTouch.



Slika 23: program DB Load

6.3 SCADA aplikacija Federal Mogul

Vhod v valjarsko linijo je jekleni trak. V procesu trak z gretjem in hlajenjem prekrijemo z bronom tako, da na izhodu linije dobimo jekleni trak z drugačnimi lastnostmi (trak je broniziran).

Linija Federal Mogul bo v prvem koraku služila za testiranje procesa broniziranja. Nameščena bo v nemškem Aachnu. Takoj ko bo testna faza zaključena (pričakuje se v približno 2 letih), se bo valjarska linija preselila v Wiesbaden. Tam jo bodo nadgradili za nadaljnje delo v proizvodnji.

SCADA aplikacija je grafični uporabniški vmesnik za uporabnika oz. operaterja, ki nadzira proizvodno linijo. Dovoljuje, zagon, zaustavitev in testiranje naprav, ki pripadajo proizvodnji liniji. Prikazuje alarme, trende, kazalnike stanja, gumbe itd.

6.3.1 Glavna pravila za SCADA aplikacijo

Za lažje razumevanje grafičnih uporabniških vmesnikov se za gradnjo uporabljajo enotna interna pravila oz. standardi, ki naredijo vmesnike tipa SCADA berljive in razumljive za uporabnika. Iz prejšnjih poglavij nam je znano, da se mora SCADA aplikacija držati določenih pravil. Tako tudi v sledeče opisani SCADI najdemo pravila, ki se jih mora načrtovalec oz. programer držati skozi celoten cikel izgradnje SCADA grafičnega uporabniškega vmesnika.

6.3.1.1 Simboli in barve

Simboli in barve imajo strateški pomen pri definiranju povezave med človekom in strojem. Glavna pravila so naslednja:

- standardne barve za specifične podsisteme ali statuse, ki poskrbijo, da operater poveže barve simbolov s statusi;

- barvanje in utripanje naredi vidni učinek glede na prioriteto prikazane informacije;
- mednarodno sprejeti standardi grafičnih simbolov za risanje in barvanje.

V tabeli 4 je vidna splošna strategija barvanja simbolov. Vsako stanje ima višjo prioriteto od predhodnega, rdeča barva se pojavi tudi, če je rumena pravkar prikazana, to pa zato, ker ima rdeča barva najvišjo prioriteto. Rdeča barva pomeni napako, medtem ko rumena napoveduje obetajočo se napako.

Tabela 4: Strategija barvanja simbolov

BARVA	STATUS
rdeča	napaka, alarm
rumena	zakasneni alarm, opozorilo
zelena	linija v obratovanju, v redu, hitrost > 0
svetlo zelena	pocasni tek, omogočeno
rožnata	lokalne kontrole
siva	stroj v redu, pripravljen

V tabeli 5 je prikazana referenca barv za cevi, kjer se pretakajo tekočine

Tabela 5: Referenca barv za cevi

TIP TEKOČINE	BARVA CEVI
olja	rjava

V tabeli 6 je prikazana referenca za barvo ventilov

Tabela 6: Referenca barv ventilov

STATUS VENTILA	BARVA SIMBOLA
zaprt	rdeča
odprt	zelena

6.3.1.2 Polja za izpis/vpis vrednosti

Pravila polja za izpis/vpis analognih vrednosti.

Izpisno/vpisna polja smo v SCADI prilagodili potrebam zaslona na dotik. To pomeni, da so polja večja in preglednejša in medsebojno zadostno odmaknjena.

Vsem vpisno/izpisnim poljem smo omejili številčno velikost vnosa in izpisa na omejitve, ki jih je predlagal naročnik projekta.

IZPISNA POLJA



Barva ozadja: bela

Barva besedila: črna

IZPISNO/VPISNA POLJA



Barva ozadja: bela

Barva besedila: modra

Slika 24: Polja za vpis in izpis podatkov

6.3.1.3 Definicije statusov in simbolov

Statusi strojev in naprav so prikazani z različnimi tipi simbolov. Ti simboli so prirejani zaslonu na dotik, so večji, preglednejši, postavljeni v določenem medsebojnem razmiku.

Statusi strojev

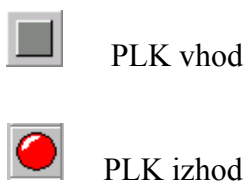
Statuse strojev smo predstavili z besedilom v obarvanem pravokotniku (slika 25). Tipična barva pravokotnika je siva, vendar smo za prikaz določenih statusov, ki so zelo pomembni, uporabili tudi rdečo in rumeno barvo (npr. napaka rdeča, alarm rumena).



Slika 25: Status stroja

Statusi paketov

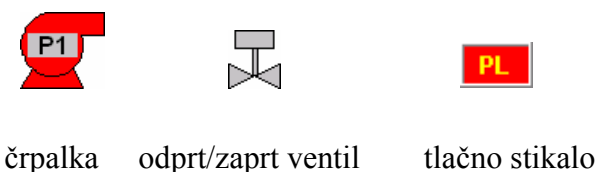
Okna, ki so namenjena paketom (strojev ne izdeluje strateški partner, ampak jih kupi v paketu s programsko opremo od drugega proizvajalca npr. Mitshubishi), so v SCADI prikazana kot aktivna komunikacija med PLK-jem in paketom. Analogne vrednosti pa so prikazane na standardni način z vpisno/izpisnimi polji (slika 26).



Slika 26: Statusi paketov

Statusi tekočin

Za cirkulacijo tekočin smo v SCADI uporabili standardne električne simbole (slika 27), ki smo jih obarvali glede na njihov status. Pri obarvanju simbolov je potrebno paziti, da so simboli obarvani v skladu s tekočino, za katero so posamezne naprave namenjene.



Slika 27: Primeri električnih simbolov

6.3.1.4 Alarmi

Namen alarmiranja je preprečevanje potencialne škode, ki bi nastala, če SCADA aplikacija ne bi vsebovala alarmov. Tako alarmi omogočijo operaterju, reakcijo (pokliče nadrejenega, vzdrževalca, ustavi proizvodni proces ipd.) ter povečajo operaterjevo budnost. Stran s povzetkom alarmov oz. alarmi v nogi SCADA so potrebni za diagnozo problema, ki sproži alarm. Po diagnosticiranju problema je operaterju na podlagi videnega omogočena ustrezna reakcija, če je le ta potrebna.

Če alarmi niso pravilno prikazani oz. ne vzbudijo operaterjeve pozornosti, so nični. Pri načrtovanju alarmov se je potrebno držati internih pravil izgradnje strani z alarmi npr. prioriteta alarmov, pravilne barvne kombinacije, vklop sirene, potrjevanje alarmov na dislociranih SCADA aplikacijah itd. ter upoštevati interna pravila grupacije alarmov.

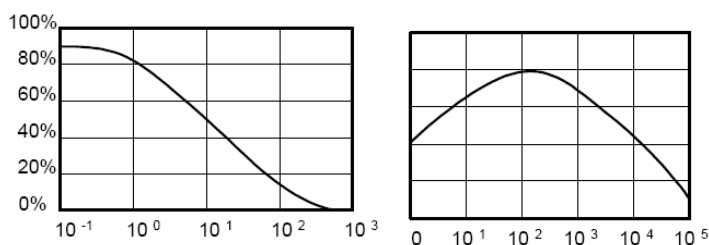
Določene barvne kombinacije (tabela 7) pri različnih prioritetah alarmov omogočajo hiter odziv operaterja, saj le ta tako hitro kot vidi določeno barvno kombinacijo, ki vzbudi njegovo pozornost nanjo tudi ustrezno reagira. V spodnji tabeli lahko vidimo, da ima alarm z visoko prioriteto barvo pisave rdečo – vklopi se sirena, ko pa operater alarm potrdi, se sirena izklopi, barva alarma se spremeni v temno sivo.

Tabela 7: Barve alarmov

Prioriteta	Opis	Barva pisave	Barva ozadja
visoka	mora biti potrjen	rdeča	črna
nizka	mora biti potrjen	rumena	črna
visoka	pravkar potrjen	temno siva	črna
nizka	pravkar potrjen	temno siva	črna
sporočila	potrditev ni potrebna	bela	črna
vse	alarmi niso aktivni, toda niso bili potrjeni	modra	črna

Alarmi so v SCADA aplikaciji grupirani zaradi množičnosti različnih alarmov, ki se lahko pojavijo. Tako je potrebno npr. grupirati alarme, ki se nahajajo na vhodu linije posebej, enako velja tudi za izhod itd.

Grupacija alarmov je dobrodošla zaradi človeškega faktorja, saj bi se npr. operater valjarske proge v množici različnih in hkrati podobnih alarmov hitro zmedel, njegovo reagiranje na alarme pa bi bilo počasnejše in neučinkovito – varnost sistema bi se bistveno zmanjšala. Na sliki 28 si lahko ogledamo učinkovitost odziva (leva slika) operaterja na število tvorjenjih alarmov na minuto. Na desni sliki pa lahko vidimo, kako niha varnost glede na število tvorjenjih alarmov.



Slika 28: Odzivnost uporabnika na alarme¹²

¹² Real Time Engineering Ltd (2006) Optimising the Human Machine Interface for Abnormal Situations, Institution of Engineering and Technology UKACC Control Conference 2006

6.3.2 Zahteve naročnika za pisanje SCADA aplikacije

Verodostojnost podatkov

Vsaka vrednost mora biti verodostojno preverjena. To pomeni, omejitve vrednosti na minimalne in maksimalne, premiki vrednosti so možni samo znotraj določenih omejenih vrednosti.

Operativnost valjarske linije

PLK mora stalno preverjati operativnost linije. To pomeni, da mora biti sposoben tvoriti prikaz vseh napak, ki se pojavijo na liniji in jih shraniti v datoteko z datumom in uro.

Nastavitve valjarske linije

Na SCADI morajo biti na posebni strani prikazani ukazi in statusi:

- normalna hitrost linije,
- napetost odvijalnika,
- napetost navijalnika,
- položaj ravnalnega valja 1,
- položaj ravnalnega valja 2,
- položaj izravnalnih valjev na vhodu,
- položaj izravnalnih valjev na izhodu.

Operativni podatki

Operativni podatki, ki morajo biti prikazani na SCADI so naslednji:

- pritisk stisljivih valjev,
- podatki o mlinu za menjavo delovnih valjev.

Vzdrževanje valjarske linije

Dostop do vzdrževalne strani na SCADA aplikaciji mora biti zaščiteno z dostopno kodo, saj je le tako onemogočimo nepooblaščenim osebam dostop do sprememb določenih zaščitnih parametrov, ki so vitalnega pomena za nemoteno delovanje valjarske linije.

Vzdrževanje mlina

Podatki na SCADA aplikaciji za vzdrževanje mlina so naslednji:

- stran operaterja/stran motorja: popravki paralelnega nastavljanja položaja,
- stran operaterja/stran motorja: umerjanje ničelnega položaja,
- obvestilo o menjavi delovnih valjev.

Varnost

Linija je obdana z varovalno ograjo. Z odpiranjem varnostnih vrat lahko operater oz. vzdrževalec dostopa do valjarske linije iz več smeri.

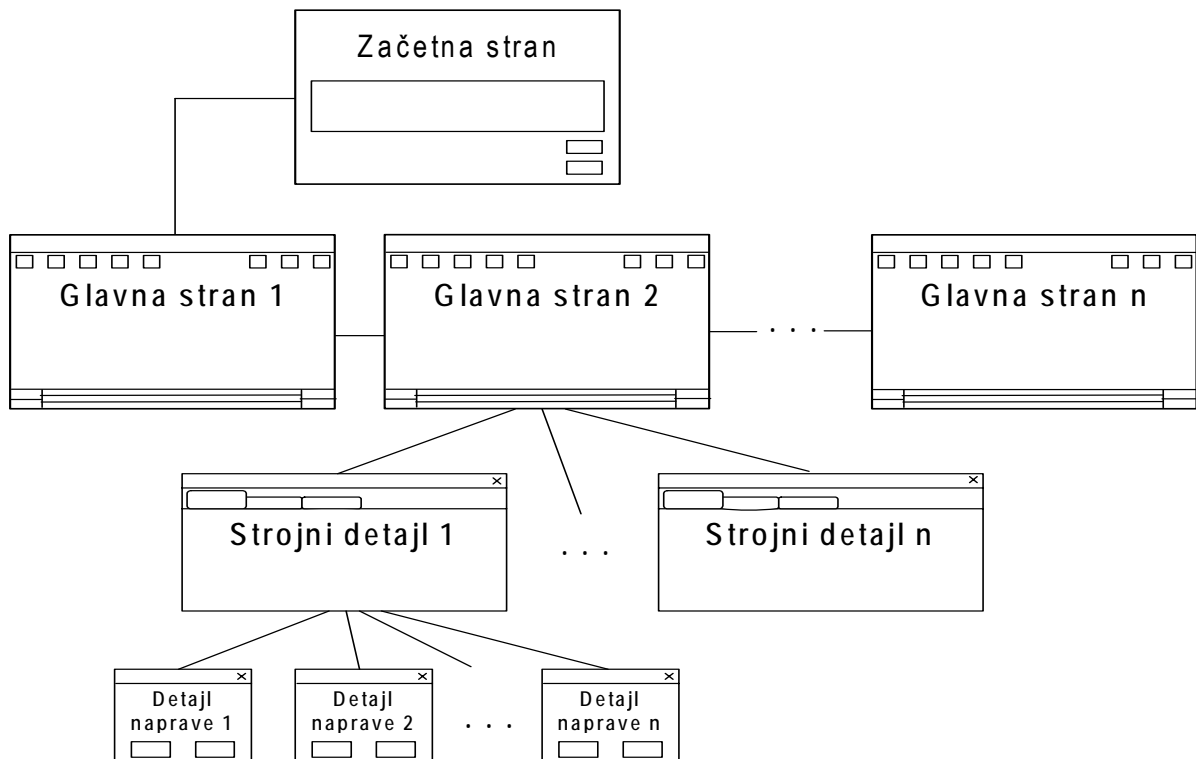
Linija je razdeljena na štiri varnostne sekcije. Razdelitev je potrebna zaradi menjave jeklenega zavitka na odvijalniku in navijalniku. Med menjavo zavitkov linija na vhodu/izhodu stoji oz. je v stanju lovljenja glave/repa traku, medtem ko sam proces teče nemoteno.

6.3.3 Prikazi strani nadzornega sistema

6.3.3.1 Organizacija strani

Prikazi na zaslonu so organizirani na več strani. Vsaka stran je v sorodu s specifičnimi podatki, ki jih vnašamo oz. kličemo iz linije.

Podatki, ki so na voljo, so prikazani v določenih oknih imenovanih tudi strani. Te strani so organizirane tako, da omogočajo celoten in strukturiran pregled nad celotno linijo (slika 29). Za PC-bazirane SCADA aplikacije so okna organizirana v nekaj glavnih strani, ki povzamejo vse poglavitne značilnosti strojev. Iz vsake glavne strani se lahko z navigacijskimi gumbi gibljemo po različnih glavnih straneh, se pravi, da se po programu premikamo horizontalno, lahko pa se pomikamo tudi vertikalno v detajle in iz detajlov določenih strojev.

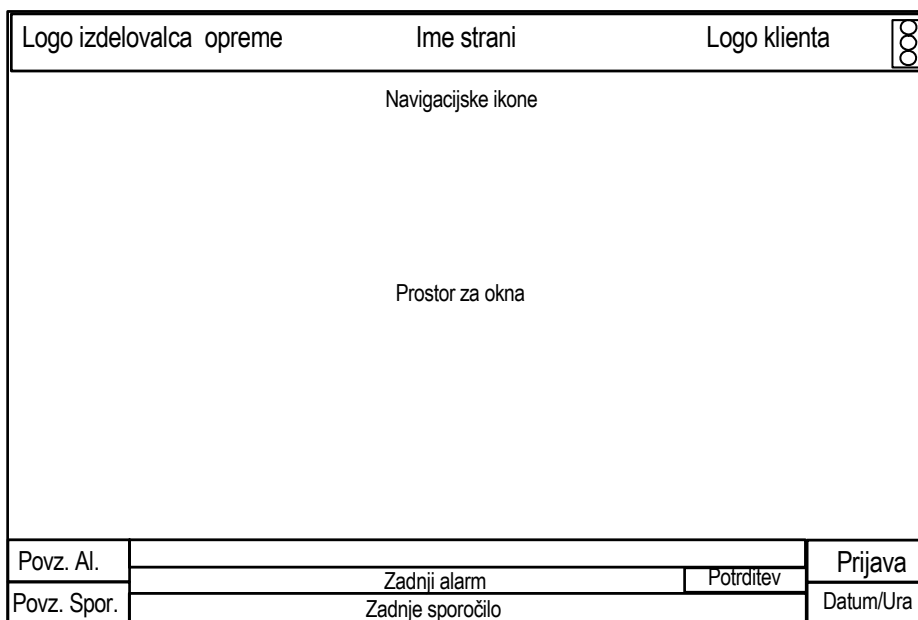


Slika 29: Organizacija strani

6.3.3.2 Tipični videz strani

Stran je razdeljena na različna podokna (slika 30), ki ponujajo naslednje:

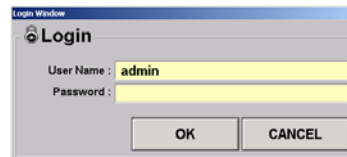
1. ime strani: pove, katero okno je aktivno;
2. datum in čas: prikaže nam enako uro in datum kot v MS Windows;
3. ikone za navigacijo: uporabljajo se za odpiranje detajlov in glavnih oken;
4. prostor za okna: prikaže nam specifične podatke;
5. povzetek alarmov in sporočil: pokliče okna s povzetki sporočil in alarmov;
6. potrditev: potrditev zadnjega alarma;
7. prijava: odpre nam »prijavno« okno.



Slika 30: Videz strani

6.3.3.3 Prijavno okno

Uporabniki se v SCADO prijavljajo z uporabniškim imenom in geslom. Na prijavnih strani (slika 31) uporabniki vnesejo dodeljeno uporabniško ime in geslo. Prijava je potrebna zaradi zaščite določenih vitalnih navigacijskih ikon, kot so npr. ikona za vzdrževanje, izhod iz SCADA idr.



Slika 31: Prijavno okno

Prijavijo se lahko kot:

- upravitelj (ima polno dostopnost funkcij na SCADA aplikaciji),
- vzdrževalec (lahko dostopa do določenih funkcij nastavitve stroja) in
- operater (operater lahko dostopa do funkcij, ki so vezane na delovanje linije).

6.3.3.4 Opis glave in noge

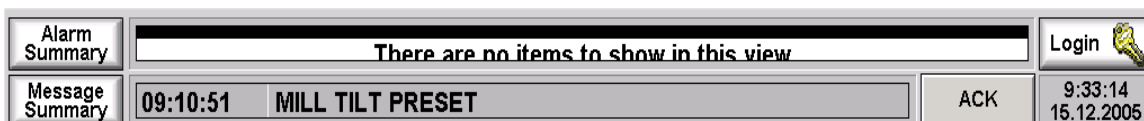
V glavi (slika 32) smo po zahtevanem standardu razporedili statute in besedila. Tukaj je prikazan emblem proizvajalca programske opreme, ime strani, semafor, ki ima sedaj obliko gumba, ter čas in datum. V glavi je najpomembnejši del semafor, saj le ta na grobo opiše delovanje linije. Rdeča luč pove, da je linija zaustavljena, ker je na njej prišlo do napake. Ta barva semaforja ima tudi najvišjo prioriteto. Rumena luč pove, da linija še vedno teče, ampak so na njej napake. Zelena luč pa pove, da linija teče brez napak. Ta barva ima tudi najnižjo prioriteto.



Slika 32: Glava SCADA aplikacije

V nogi smo ravno tako kot v glavi postavili vse statuse, gumbe in besedila po zahtevanem standardu.

Gumbi in besedila so opisani v tipičnem videzu strani. V nogi SCADA lahko vidimo tudi izpise zadnjih alarmov in zadnjih sporočil. Ne glede na prioriteto alarmov in sporočil morajo biti vsi alarmi potrjeni. S potrditvijo operater linije alarm na SCADI označi kot opažen. Sporočila nimajo potrebe po potrjevanju. Če pa operaterja zanima celoten pregled vseh alarmov oz. sporočil, mu tipki Alarm Summary in Message Summary omogočata celoten pregled nad zgodovino alarmov oz. sporočil (slika 33).



Slika 33: Noga SCADA aplikacije

6.3.3.5 Navigacijske ikone

Navigacijske ikone (slika 34) so na zgornjem delu, takoj pod glavo SCADA aplikacije. Na desni strani vidimo ikone, ki se uporabljajo za navigacijo po generalnih straneh (vhod, izhod, celotne linije). Ikone na levi strani pa prikazujejo vse dostopne podatke o nastavitvah mreže, napetosti traku, varnostnih vratih, motorjih, hidravliki, nastavitvah strojev in vzdrževanju. Določene ikone leve strani se ob napakah obarvajo rdeče.



Slika 34: Navigacijske ikone

6.3.3.6 Prikaz procesnih strani

Na procesnih straneh so prikazani najvitalnejši podatki za pravilno delovanje linije. Prikazani so tisti statusi in ukazi, ki jih je naročnik SCADA aplikacije zahteval v svojih

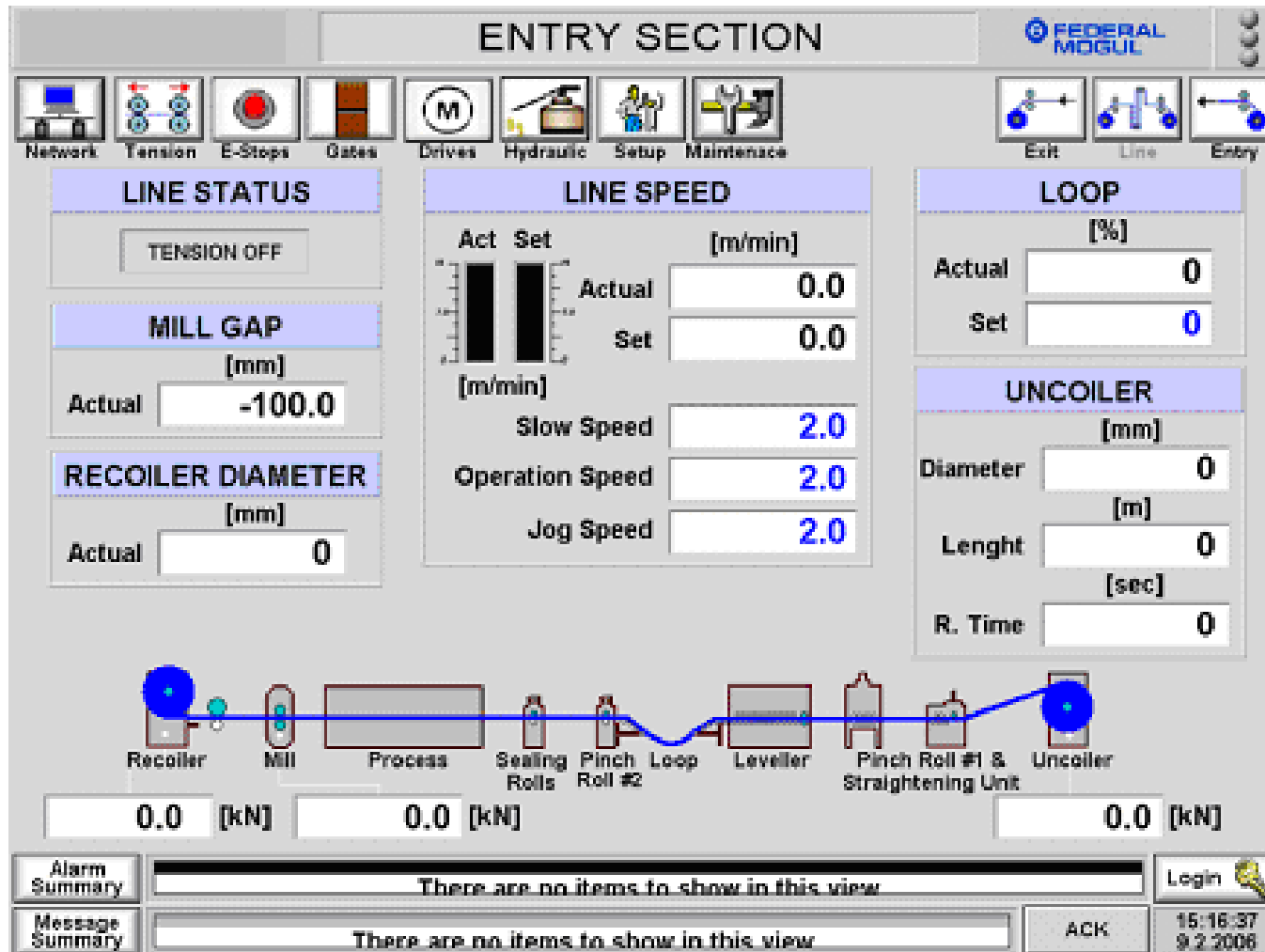
zahtevah. Glavne procesne strani so razdeljene na tri dele, in sicer na vhodno sekcijo, celotno linijo in izhodno sekcijo.

Pregled celotne linije

Procesna stran, ki prikazuje celotno linijo (slika 35), vsebuje naslednje statuse in ukaze:

- status linije, ki pove, ali je linija v prostem teku ali v stanju delovanja pod napetostjo jeklenega traku (silo napetosti merimo na odvijalniku, navijalniku ter mlinu);
- hitrost celotne linije, ki jo lahko poljubno spreminjamo z določenimi številskimi omejitvami (linija ne more biti hitrejša od maksimalno določene hitrosti);
- statuse, ki se nanašajo na premer, dolžino, odvijalni čas jeklenega zavitka, na odvijalniku in navijalniku;
- statusi in ukazi procentualne napolnjenosti akumulatorja;
- razdaljo med delovnimi valji v mlinu.

Na sliki 35 vidimo celotno valjarsko linijo s poudarjenim vhodom in izhodom linije, saj zahteve za podrobnosti procesa niso znane.



Slika 35: Pregled celotne linije

Vhod valjarske linije

Vhodna sekcija valjarske linije (slika 36) vsebuje opis vsakega posameznega stroja v tej sekciji. Vidimo lahko osnovne statute in ukaze, ki so pomembni za vsak stroj posebej. Po navadi so na takih in podobnih straneh statusi, ki se tičejo razmika med valji stroja. Tako npr. ta stran vsebuje statute, ki prikazujejo milimetrsko razdaljo med delovnimi valji. Najdemo pa tudi ukaze za prosti tek strojev, ki se nanašajo na lovljenje glave traku.

Operater podrobne ukaze in statute najde na posebnih straneh, do katerih dostopamo s pritiskom na sliko posameznega stroja. Ti ukazi in statusi se v praksi malo uporabljajo, zato jih je bilo nesmiselno postavljati po procesnih staneh, saj bi tako izgubili preglednost SCADA aplikacije.

Če dobro pogledamo vidimo, da so nekateri statusi in ukazi enaki v celotnem pregledu linije. Enaki so zaradi njihove pomembnosti in pogostosti rabe v procesu valjanja, tako da operaterju ni potrebno premikanje po ostalih procesnih oknih, da bi videl osnovne statute in ukaze.

ENTRY SECTION

Network
 Tension
 E-Steps
 Gates
 Drives
 Hydraulic
 Setup
 Maintenance

Exit
 Line
 Entry

LOOP	LEVELLER GAP	STRAIGH. UNIT	COIL DIAMETER
[%]	[mm]	[mm]	[mm]
Actual <input type="text" value="0"/>	Entry <input type="text" value="-100.00"/>	P. #1 <input type="text" value="-100.00"/>	Actual <input type="text" value="0"/>
Set <input type="text" value="0"/>	Exit <input type="text" value="-100.00"/>	P. #2 <input type="text" value="-100.00"/>	

Alarm Summary

There are no items to show in this view

Login

Message Summary

There are no items to show in this view

ACK

17:20:15
9.2.2008

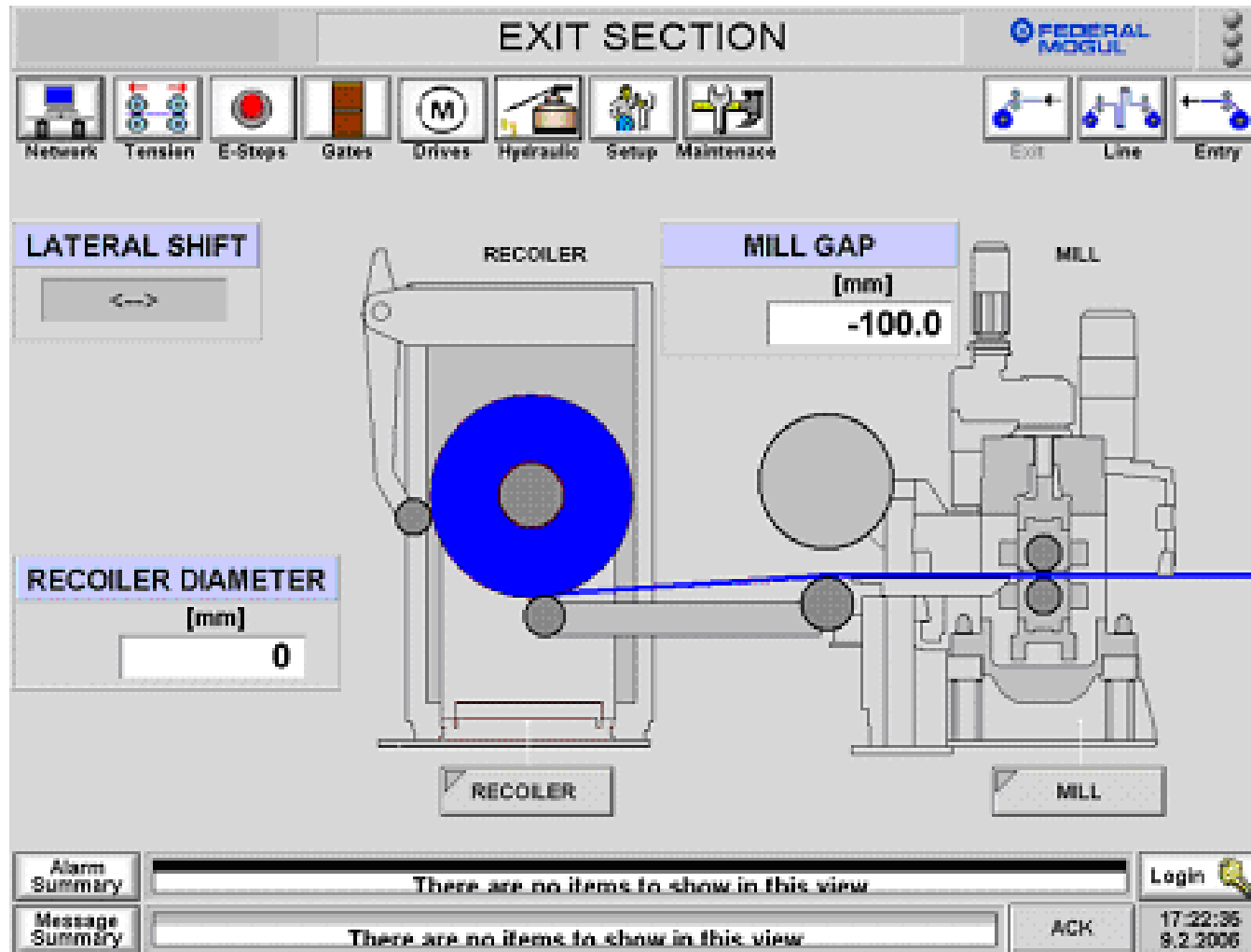
Slika 36: Pregled vhoda linije na SCADA aplikaciji

Izhod valjarske linije

Izhodna sekcija valjarske linije (slika 37) vsebuje opis vsakega posameznega stroja, ki je v tej sekciji. Tu vidimo osnovne statuse in ukaze, ki so pomembni za vsak stroj posebej. Navadno so na takih in podobnih straneh statusi, ki se tičejo razmika med valji določenega stroja. Tako npr. stran vsebuje statuse, ki prikazujejo milimetrsko razdaljo med delovnimi valji mlina. Na spodnjem delu slike 37 vidimo tudi ukaze za počasni tek strojev (ang. jog), ki se nanašajo na lovljenje glave traku.

Operater podrobne ukaze in statuse najde na posebnih straneh, do katerih dostopamo s pritiskom na sliko posameznega stroja. Ti ukazi in statusi se v praksi malo uporabljajo, vendar so npr. za vzdrževanje mlina zelo pomembni. Ker se mlin ne vzdržuje vsak dan, jih je bilo nesmotrno postavljati na glavno izhodno procesno stran.

Tu vidimo, da so nekateri statusi in ukazi enaki kot v celotnem pregledu linije. Taki so zaradi svoje pomembnosti in pogostosti rabe v procesu valjanja, tako da operaterju ni potrebno premikanje po ostalih procesnih oknih, da bi videl osnovne statuse in ukaze.



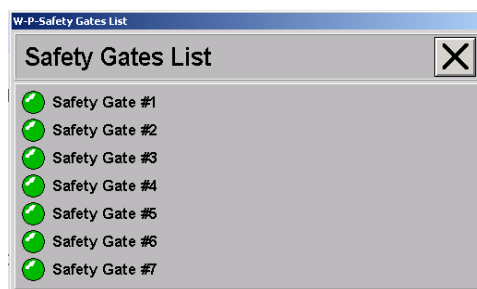
Slika 37: Pregled izhoda linije

6.3.3.7 Varnostna vrata

Statuse varnostnih vrat smo izvedli s pojavnim oknom (ang. Pop-up). Na njem (slika 38) lahko vidimo seznam in statuse vrat, ki so na vseh varnostnih sekcijah. Za prikaz stanja smo uporabili indikatorske lučke, ki se obarvajo glede na stanje vrat.

Barve so slednje:

- zelena označi, da so vsa vrata zaprta – linija je v teku;
- rdeča opozori, da so vrata odprta – linija je ustavljena;
- rumena potrdi, da je PLK dobil zahtevo za odpiranje vrat.



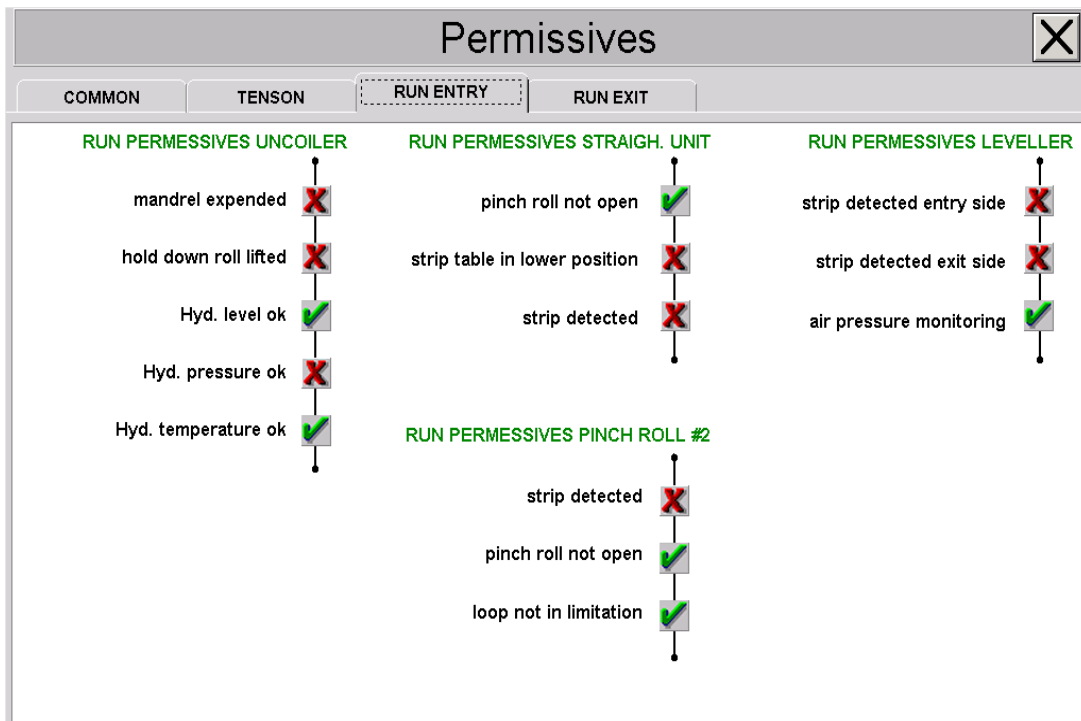
Slika 38: Lista varnostnih vrat

6.3.3.8 Dovoljenja za delovanje linije

SCADA aplikacija vsebuje poseben skupek strani (slika 39), na katerih si operater ogleda dovoljenja, ki morajo biti prisotna za zagon linije (premik jeklenega traku) in vzdrževanje funkcije napetosti.

Dovoljenja za vzdrževanje strojev v funkciji napetosti so potrebna za zagon linije. Ko je linija v delovanju, morajo biti vsa dovoljenja v logičnem stanju »true« ves čas delovanja.

Dovoljenja »za delovanje« so potrebna za premike traku skozi valjarsko linijo. Če dovoljenje spremeni stanje bita iz logičnega stanja »true« na »false« se premikanje traku ustavi. To pa še ne pomeni, da je linija izgubila dovoljenje za funkcijo napetosti traku.



Slika 39: Skupek strani dovoljenj

6.3.3.9 Alarmi

V tabeli 8 vidimo grupiranje skupnih področij alarmov. Alarme grupiramo po sorodnosti glede na tip. Tako npr. skupaj grupiramo alarme, ki se nanašajo na hidravliko linije, na vhod, izhod linije ipd. Vnos alarmov v procesno bazo poteka na enak način kot je predhodno opisano v poglavju 6.2.1.1.

Tabela 8: Primer Excelove tabele v kateri so definirani alarmi

NASLOV	IME NALOGE	OPIS	SPR.	TIP
DBX 000.00	LNCOM_BA_AUX_24_FLT	Auxiliary 24V Fault	BB	AG000
DBX 000.01	LNCOM_BA_SOL_24_FLT	Solenoid Valves 24V Fault	BB	AG000
DBX 001.00	LNCOM_BA_CPU_BAT_ALM	Line PLK Battery Fault	BB	AG000

Na sliki 40 vidimo tabelo alarmov izvedeno v InTouchu. V področje alarmov spada tabela alarmov, ki se podobno kot tekstovni izpis glede na bit izpisujejo na zaslon. Glede na grupiranje alarmnega področja, ki smo ga predhodno definirali ter opisali, se izpisujejo alarmi iz tabele 8. Primer: alarm z določenim komentarjem se bo izpisal, ko bo bit DB 284, DBX 000.00 postavljen na logično enico oz. bo pridobil logično vrednost »true«. PLK signalizira natančno napako in s tem čas za iskanje le-te praktično odpade. Vsi alarmi se shranijo v datoteko z datumom in uro, tako je možen kasnejši vpogled ter določitev časa ter vzroka zastoja. Omogočeno je določeno manipuliranje z alarmi npr.: glej vse alarme, glej potrjene alarme, glej nepotrjene alarme ipd, pa tudi izpisovanje alarmov na papir.

EXIT SECTION

FEDERAL
MOGUL

Alarm Summary

AREA / GROUP: **(AG000)** X

Date	Time	Comment	
15/12/05	14:50:50	MAX TILTING REACHED	KNOWLEDGE <input type="button" value="ACK. ALL"/> <input type="button" value="DISPLAYED"/> <input type="button" value="SELECTED"/> <input type="button" value="VIEW FILTER"/> <input style="background-color: green; color: white;" type="button" value="VIEW ALL"/> <input type="button" value="VIEW ACK."/> <input type="button" value="VIEW UNACK."/>
15/12/05	14:48:55	DS MAX CLOSED POSITION	
15/12/05	14:30:25	OS MAX CLOSED POSITION	
15/12/05	10:21:24	OS MAX OPENED POSITION	
15/12/05	09:33:24	DS MAX OPENED POSITION	
15/12/05	09:06:33	=RP01E00EMP01 UNCOILING EMERGENCY STOP	ALARM TOTAL <div style="border: 1px solid black; padding: 2px; text-align: center; font-weight: bold; font-size: 1.2em;">11</div> PAGE <div style="border: 1px solid black; padding: 2px; text-align: center; font-weight: bold; font-size: 1.2em;">1/1</div> <input type="button" value="PRINT"/>
15/12/05	09:06:33	=RP01E00EMP01 UNCOILING EMERGENCY STOP	
15/12/05	09:06:33	=RP01E00EMP01 LINE EMERGENCY STOP	
15/12/05	09:06:33	=RP01E00EMP01 LINE EMERGENCY STOP	
15/12/05	09:06:33	=RP01E00EMP01 RECOILING EMERGENCY STOP	
15/12/05	09:06:33	=RP01E00EMP01 RECOILING EMERGENCY STOP	

Update Successful

Alarm Summary	14:50:59	MAX TILTING REACHED	Login
Message Summary	15:07:29	Pay-Off Hydraulic Pump - Working Hours RESET	ACK 15:11:32 15.12.2005

Slika 40: Stran skupnih alarmov

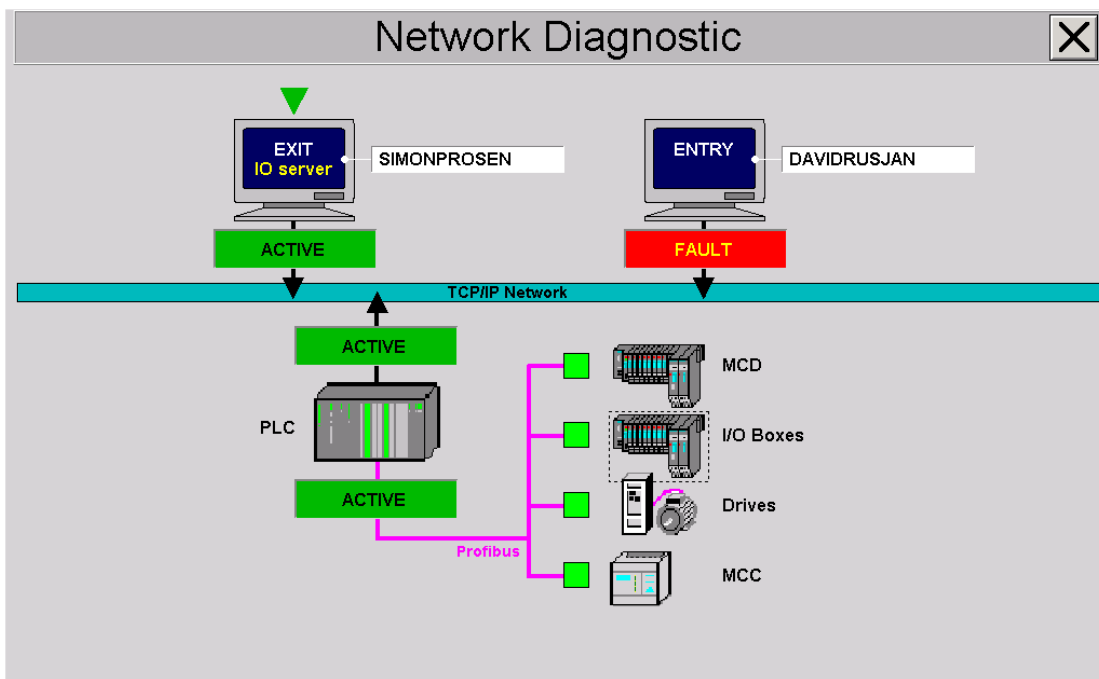
6.3.3.10 Diagnostika mreže

Stran za diagnostiko mreže (slika 41) je uporabna za reševanje problemov s komunikacijo. Iz same slike lahko razberemo, da gre za porazdeljen računalniški sistem, vidimo, da valjarska linija uporablja dva osebna računalnika (PC-ja), ki sta na vhodu in izhodu linije (višji nivo). Mreža je zgrajena po načelu tipologije vodila. Osebna računalnika sta medsebojno povezana preko TCP/IP (ang. Transfer Control Protocol/Internet protocol) protokola, na katerega je vezan tudi PLK (nižji nivo). Le ta pa komunicira z napravami, kot so motorji, vhodno/izhodne omarice itd. preko vodila Profibus (ang. Professional bus). Proces obdelave podatkov poteka na nižjem nivoju hitreje kot na višjem. Naloge posameznega nivoja so opisane v spodnji tabeli (tabela 9):

Tabela 9: Izvedba nalog po nivojih

NIVO	NALOGE	OPREMA
Nižji	izvedba regulacijskih zank, vodenje naprav procesa valjanja, lokalna kontrola	PLK Simatic 400, aktuatorji, senzorji
Višji	komunikacija med človekom in strojem, arhiviranje podatkov, alarmiranje, pomoč pri odločanju	Osebni računalnik, SCADA aplikacija

V praksi operater vidi samo dva ekrana, ki sta občutljiva na dotik (eden na vhodu, drugi na izhodu).



Slika 41: Diagnostika mreže

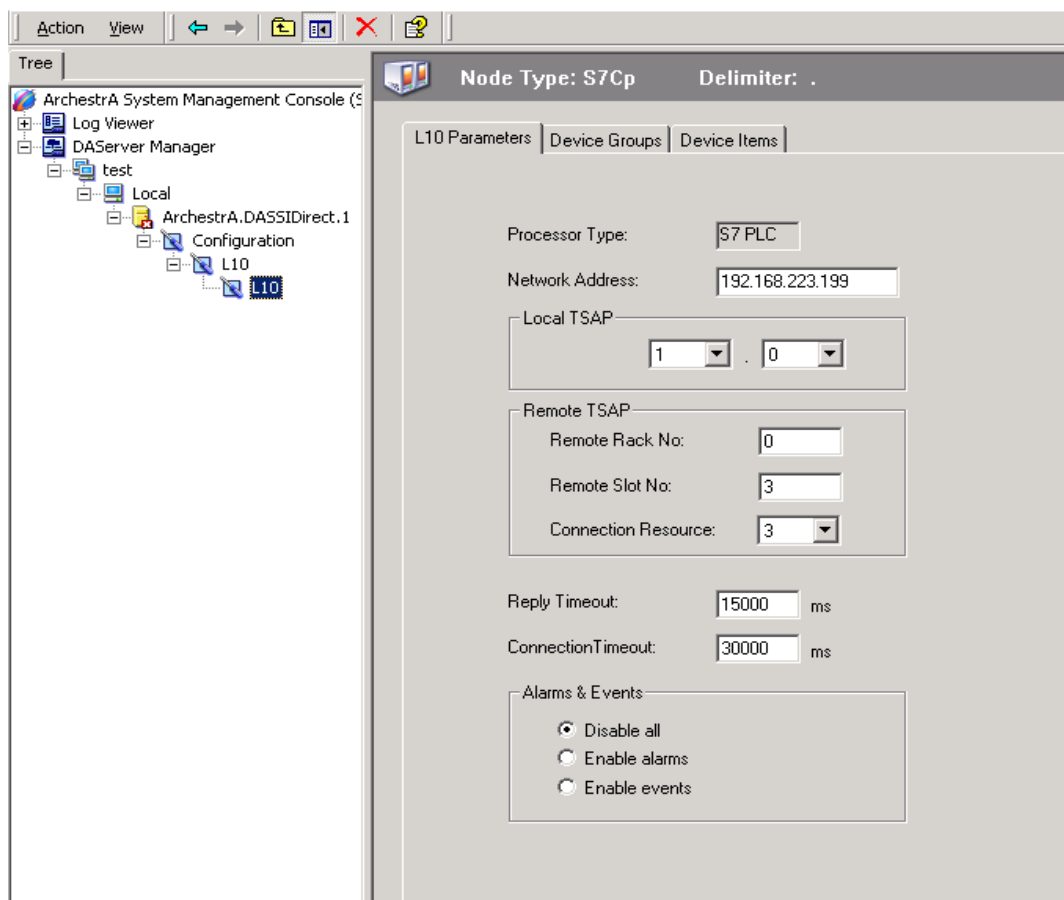
V delovanju SCADA komunicira direktno s PLK-jem preko vhodno/izhodnega strežnika, ki se obnaša kot strežnik za druge SCADA aplikacije. Besedilo na SCADI pove, kateri Vhodno/izhodni strežnik je aktiven. Možna je tudi zamenjava strežnika z drugo SCADA aplikacijo.

Tako se lahko na SCADA aplikaciji ločeno prikazuje podatke o liniji. To pomeni, da operater na vhodu linije ne more videti delovnih parametrov, ki so na izhodu linije in obratno. Nadzor linije iz katerekoli SCADA pa je omogočen upravitelju. Operaterju je omogočen pregled nad celotno linijo le takrat, ko se pojavi napaka pri komunikaciji med SCADA aplikacijama, ki privede do nepravilnega delovanja ene izmed dveh SCADA aplikacij.

6.3.4 Postavitev krmilnika med SCADO in PLK-jem

Komunikacija med SCADA aplikacijo in PLK-jem poteka preko programa Dassidirect. Konfiguracijo krmilnika med PLK-jem in SCADO, izvajamo z zunanjim Wonderware programom, imenovanim Orchestra System Management Console (slika 42). Postopek je naslednji:

- vpisati je potrebno mrežni naslov oddaljenega PLK-ja (PLK IP);
- popraviti je potrebno Remote Tsap, ki se spreminja glede na konfiguracijo PLK-ja;
- zažene komunikacijski program DassiDirect.



Slika 42: Nastavitve komunikacije med SCADO in PLK-jem

6.3.5 Preizkus SCADA Federal Mogul

SCADA je bila »hladno« preizkušena na sedežu podjetja in strateškega partnerja v Butriu. Testiranje je bilo izvedeno po naslednjem postopku:

- Vzpostavitev komunikacije SCADA-PLK preko programa Dassidirecta v Achestri. Komunikacija je vzpostavljena preko Etherneta na PLK Simens 400.
- Komunikacija mora biti vzpostavljena tudi med SimaticNetom in PLK-jem.
- V PLK je predhodno preko Etherneta naložena aplikacija SimaticNet, ki krmili PLK.
- Po vzpostavitvi komunikacije med elementi sledi postavljanje bitov na logično vrednost »true« ali »false« oz. »0« ali »1« preko lestvičnega diagrama. Vrednost bitov, ki se nanašajo na statute ali alarme, smo spreminjali z vzpostavljanjem »obvodnic«, vrednosti ukazov oz. ukaznih gumbov pa s kontrolo signala na izhodu izbranega bloka v SimaticNetu.

Preizkušanje je potekalo približno teden dni. V tem času je bilo potrebno preizkusiti vse statute, ukaze ter alarme. Če pa se bodo dodatno pri zagonu pojavile še kakšne napake, se bodo odpravile v času samega zagona. Pomoč pri popravkih na terenih se vodi iz matičnega podjetja preko elektronske pošte oz. telefona.

7 ZAKLJUČEK

Diplomsko nalogo smo razčlenili na pet sklopov. Bralci ste verjetno opazili, da se diplomsko delo dotika vseh parametrov, ki so potrebni za uspešno izvedbo grafičnega uporabniškega vmesnika. Če bi eden od teh sklopov manjkal, uspešnost realizacije - v našem primeru SCADA aplikacija, ne bi dosegla primerne ravni uporabnosti in izgleda.

Kot vemo, potrebuje vsak praktični oz. realni izdelek neko teoretično osnovo, zato v tretjem sklopu razlagamo teoretske osnove izgradnje grafičnih uporabniških vmesnikov. Ugotovili smo, da tukaj ni nekega splošnega pravila, ki bi nas vodil do zastavljenega cilja, ampak je treba predvsem upoštevati želje kupca ter zmožnosti današnje tehnologije.

V okviru diplomske naloge je bilo potrebno zgraditi uporabniški vmesnik nadzornega sistema oz. SCADA aplikacijo za valjarsko linijo Federal Mogul. V poglavju o SCADA aplikaciji so opisani notranji standardi, zahteve naročnika linije za pisanje SCADA ter naše rešitve za dodeljene naloge.

Naloge so bile, kot se je izkazalo na preliminarnem preizkusu, opravljene uspešno. Lahko pa se pojavijo določena odstopanja na samem terenu, zaradi pomanjkanja komunikacije in dokumentacije med naročnikom, prejemnikom in izvajalcem naročila. Take probleme se ponavadi rešuje naknadno.

Po zaključeni preizkusni fazi procesa v nemškem Achnu, se bo linija nadgradila za nadaljnjo proizvodnjo v Weisbadnu. Ker je bila naša naloga avtomatizirati samo vhod in izhod linije, obstaja še možnost proučitve dodatne avtomatizacije procesne linije.

8 LITERATURA

Dan R. Olsen, Jr (1998) Developing User Interfaces. San Francisco, Kalifornija: Morgan Kaufmann Publishers, inc.

FERI – Maribor. Pridobljeno 21.2.2006 s svetovnega spleta: http://lpa.feri.uni-mb.si/Pedagosko_delo/Snovanje_sistemov_vodenja/2_teden.pdf

Franc Jager, Uporabniški vmesniki, Interno gradivo FRI. Pridobljeno 29.6.2006 s svetovnega spleta: <http://www.e-student.si/FRI:VSP:UV:HitriZapiski:2006-02-24:Igors-18k->

Mojca Kralj (2005) Politike tehnološkega razvoja avtomobilske industrije, magistrsko delo, Fakulteta za družbene vede Univerze v Ljubljani: [M. Kralj]

NEC/MITSUBISHI, Touch panel technology. Pridobljeno 29.6.2006 s svetovnega spleta: http://www.necdisplay.com/support/css/Techlibrary/whitepaper_touchtechnology.pdf

Nikolaj Pečenko (2006) Okna, ikone, meniji & miške, Monitor, januar 2006.

Real Time Engeneering Ltd (2006) Optimising the Human Machine Interface for Abnormal Situations, Institution of Engeneering and Technology UKACC Control Conference,

S. Strmčnik (ur.) (1998). Celostni pristop k računalniškemu vodenju procesov Fakulteta za elektrotehniko

Wnderware InTouch. Pridobljeno 21.2.2006 s svetovnega spleta: <http://www.wweng.com/html/produkti.html>